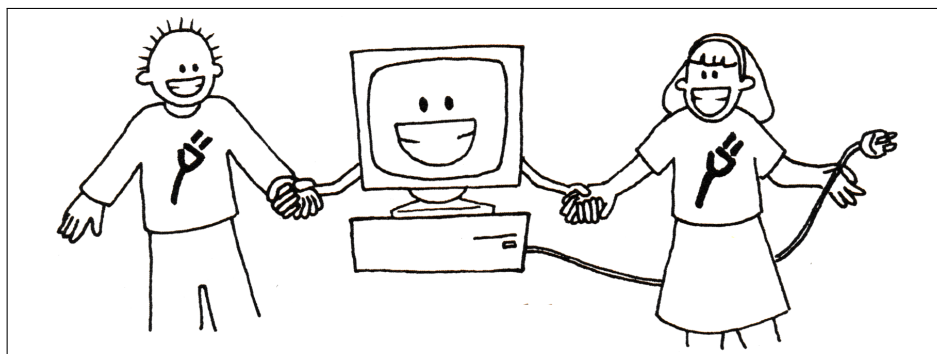


CS UNPLUGGED



**Imparare l'informatica senza alcun Computer.
Una collezione di attività didattiche divertenti per gli
studenti delle scuole primarie**



Creato da:

Tim Bell, Ian H. Witten and Mike Fellows

Adattato per l'uso in classe da Robyn Adams e Jane McKenzie

Illustrazioni di Matt Powell

Revisione del 2015 di Sam Jarman

Edizione Italiana a cura di:

Renzo Davoli, Giovanni Michele Bianco, Piergiovanna Grossi

Revisione 2015. Versione 1.0.1.

This work is licensed under a Creative Commons "Attribution-NonCommercial-ShareAlike 3.0 Unported" license.



Introduzione

I Computer sono ovunque. Tutti noi dobbiamo sapere come usarli e molti di noi li usano tutti i giorni. Ma come funzionano? Come pensano? E come si fa a renderli più veloci e affidabili, insomma migliori? L'informatica è una scienza affascinante che risponde a queste domande. Le semplici e divertenti attività di questo libro, adatte a bambini e ragazzi di diverse età, introducono i concetti fondamentali dell'informatica, senza che gli studenti debbano usare alcun computer.

Questo libro può efficacemente essere usato in programmi di approfondimento o anche durante le normali lezioni. Non dovete essere esperti di computer per potervi divertire nell'imparare i principi dell'informatica coi vostri allievi. Il libro comprende diverse attività, con chiare e semplici spiegazioni e risposte alle domande. Ogni attività si conclude con un capitolo denominato "cosa c'entra tutto questo?" che spiega la rilevanza delle attività.

Molte delle attività sono correlate ad argomenti di matematica, per esempio l'esplorazione dei numeri binari, mappe e grafi, problemi di riconoscimento e di ordinamento, crittografia. Altre attività riguardano argomenti di solito trattati in corsi di tecnologia, come per esempio l'apprendimento di come effettivamente funziona un computer. Gli studenti sono coinvolti in attività che sviluppano le capacità di risoluzione di problemi, di comunicazione e la creatività in un contesto significativo ma anche divertente. Le attività presentate in questo libro forniscono un modo molto avvincente di esplorare il pensiero computazionale, concetto che assume sempre più rilevanza nei programmi scolastici.

Oltre a questo libro, il progetto Unplugged fornisce tante risorse gratuite online quali video, immagini e materiale didattico all'indirizzo csunplugged.org. Come attività correlata alla edizione 2015 del libro è stato realizzato un nuovissimo sito web, con più risorse, migliore accesso al materiale open-source e indicazioni per poter inserire l'informatica e il pensiero computazionale all'interno dei piani formativi scolastici. (N.d.T. Il nuovo sito web è in lingua inglese).

Questo libro è stato scritto da tre professori universitari di Informatica e da due insegnanti di scuola ed è basato sulle esperienze dirette in classe e sui riscontri e commenti di centinaia di educatori che usano questo programma didattico da oltre due decenni. Abbiamo visto che molti importanti concetti di Informatica possono essere insegnati senza un computer. Talvolta un computer è al contrario una distrazione che

riduce la capacità di apprendimento. Quindi scollegate (unplug!) i vostri computer e preparatevi ad imparare ciò che veramente è l'Informatica.

Questo libro è disponibile per il download gratuito grazie a una generosa donazione di Google, Inc. È distribuito secondo le regole della licenza Creative Commons Attribuzione – Non Commerciale – Condividi allo stesso modo, che significa che siete liberi di copiare, distribuire, trasmettere il libro. È anche permesso modificare il libro. Tutto ciò è consentito a patto che vengano sempre citati gli autori dell'opera, che l'opera non venga usata per scopi commerciali e che se l'opera viene modificata, trasformata e rielaborata, il risultato venga sempre rilasciato con la stessa licenza. Chi fosse interessato ai dettagli della licenza può cercare sul web CC BY-NC-SA 3.0.

Noi incoraggiamo l'uso di questo testo per scopi didattici e per questo fine siete autorizzati a stampare la vostra copia del libro e a distribuire i fogli di lavoro ai vostri studenti. Tutte le richieste e i suggerimenti sono benvenuti, potete inviarli direttamente agli autori come indicato nel sito www.unplugged.org.

Questo libro è stato tradotto in molte lingue. Potete verificare nel sito web la disponibilità delle traduzioni del testo.

Ringraziamenti

Molti ragazzi e molti insegnanti ci hanno aiutato a perfezionare le nostre idee. Gli studenti e gli insegnanti della South Park School (Victoria, BC), Shirley Primary School, Ilam Primary School e della Westburn Primary School (Christchurch, New Zealand) sono state le nostre cavie per molte attività. Siamo particolarmente grati a Linda Picciotto, Karen Able, Bryon Porteous, Paul Cathro, Tracy Harrold, Simone Tanoa, Lorraine Woodfield e Lynn Atkinson per averci accolto nelle loro classi e per i loro preziosi suggerimenti per il perfezionamento delle attività. Gwenda Bensemman ha sperimentato molte delle attività per noi e ci ha suggerito modifiche. Richard Lynders e Sumant Murugesh ci hanno aiutato nei tentativi in classe. Parte delle attività di crittografia sono state sviluppate da Ken Noblitz. Alcune attività sono state portate avanti sotto l'ombrello del gruppo denominato Victoria "Mathmania", con l'aiuto di Kathy Beveridge. Le prime versioni delle illustrazioni sono state fatte da Malcolm Robinson e Gail Williams e ci siamo anche avvalsi dei consigli di Hans Knutson. Matt Powell ha anche fornito una preziosa assistenza durante lo sviluppo del progetto "Unplugged". Siamo grati alla Brian Mason Scientific e alla Technical Trust per i generosi contributi durante le prime fasi di sviluppo di questo libro.

Un ringraziamento speciale va a Paul e Ruth Ellen Howard, che hanno sperimentato molte delle attività e fornito numerosi utili suggerimenti. Anche Peter Henderson, Bruce McKenzie, Joan Mitchell, Nancy Walker-Mitchell, Gwen Stark, Tony Smith, Tim A. H. Bell¹, Mike Hallett e Harold Thimbleby hanno fornito molti utili suggerimenti.

Abbiamo un enorme debito con le nostre famiglie: Bruce, Fran, Grant, Judith e Pam per il loro supporto e Andrew, Anna, Hannah, Max, Michael e Nikki che ci hanno ispirato una grande parte di questo lavoro² e che spesso sono stati i primi a sperimentare le attività.

Siamo particolarmente riconoscenti a Google Inc. per aver finanziato il progetto Unplugged e per averci consentito di realizzare questo testo disponibile per l'uso gratuito.

Tutte le richieste e i suggerimenti sono benvenuti, potete inviarli direttamente agli autori come indicato nel sito www.unplugged.org.

¹Non ci sono relazioni di parentela col primo autore.

²Infatti l'attività relativa alla compressione del testo è stata inventata da Michael.

Indice

Introduzione	i
Ringraziamenti	iii
I dati: la materia prima per rappresentare le informazioni	3
Conta i punti — <i>I numeri binari</i>	5
Colorare con i numeri — <i>La rappresentazione delle immagini</i>	23
Puoi dirlo nuovamente! — <i>La compressione del testo</i>	35
La magia delle carte girate — <i>Il riconoscimento e la correzione degli errori</i>	45
Indovina indovinello — <i>La Teoria dell'Informazione</i>	57
Far lavorare i computer — gli Algoritmi	69
Battaglia navale — <i>Algoritmi di ricerca</i>	71
Il piccolo e il grande — <i>Algoritmi di ordinamento</i>	93
Batti il tempo — <i>Reti di ordinamento</i>	103
La città fangosa — <i>Minimal Spanning Tree</i>	111
Il gioco dell'arancia — <i>Instradamento e deadlock nelle reti.</i>	117
Le Tavole di Pietra — <i>Protocolli di comunicazione</i>	121
Dire ai computer cosa devono fare — Rappresentare le procedure	131
Caccia al tesoro — <i>Automati a stati finiti</i>	133
Gli ordini di marcia — <i>I linguaggi di programmazione</i>	153
Problemi veramente difficili — Intrattabilità	161
Il cartografo povero — <i>La colorazione dei grafi</i>	163
La città turistica — <i>Gli insiemi dominanti</i>	179
Strade ghiacciate — <i>Gli alberi di Steiner</i>	189
Condividere segreti e combattere il crimine — la Crittografia	205
Condividere i segreti — <i>Protocolli che nascondono l'informazione</i>	209
Testa o croce in Perù — <i>I protocolli crittografici</i>	215
Kid Krypto — <i>La crittografia a chiave pubblica</i>	229
Il volto umano dell'elaborazione — Interagire con i computer	243
La fabbrica di cioccolato — <i>Progettare l'interfaccia utente</i>	247
Conversazioni coi computer — <i>Il test di Turing</i>	263

Parte I

**I dati: la materia prima per
*rappresentare le informazioni***

I dati: la materia prima

Come vengono immagazzinate le informazioni nei Computer?

La parola "computer" viene dal latino *computare*, che significa calcolare o sommare insieme, ma i computer di oggi sono molto di più di giganti calcolatrici! Possono essere biblioteche, ci aiutano a scrivere, a cercare informazioni, ci fanno ascoltare brani musicali o vedere film. Ma come fanno a immagazzinare, a memorizzare, tutte queste informazioni? Che ci crediate o no, il computer usa solo due oggetti: lo zero e l'uno.

Qual è la differenza fra dati e informazioni?

I dati sono materie prime grezze, i numeri con i quali il computer lavora. Un computer trasforma i dati in informazioni (parole, numeri, immagini, suoni) che potete comprendere.

Come si possono convertire numeri, lettere, parole, figure, suoni in zero e uno?

In questa parte del libro impareremo cosa sono i numeri binari, come i computer disegnano le figure, come funziona un fax, quale sia il modo efficiente per memorizzare tanti dati, come evitare gli errori di memorizzazione e come possiamo misurare la quantità di informazioni che tentiamo di immagazzinare.



Attività 1

Conta i punti — *I numeri binari*

Sommario

I computer rappresentano e trasferiscono i dati come sequenze di zero e uno. Come possiamo rappresentare parole e numeri usando solamente questi due simboli?

Competenze richieste:

Gli studenti devono essere in grado di:

- ✓ Contare
- ✓ Creare corrispondenze
- ✓ Mettere in sequenza

Età

- ✓ A partire da 6 anni

Materiale

- ✓ Serve un set di cinque carte (vedi pagina 11) per la spiegazione. In alternativa possono essere utilizzati fogli A4 con adesivi di smiley (faccine sorridenti) come punti.

Ogni studente deve avere:

- ✓ Un set di cinque carte. (Fotocopiate pagina 11 su di un cartoncino e ritagliate le carte).
- ✓ Il foglio di lavoro “numeri binari” (pagina 9)

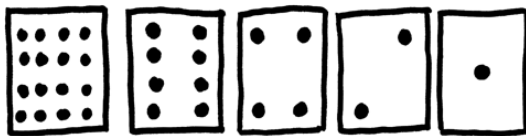
Sono previste anche attività di approfondimento opzionali, per le quali ogni studente deve avere:

- ✓ Il foglio di lavoro: “lavorare con i numeri binari” (pagina 13)
- ✓ Il foglio di lavoro: “spedire messaggi segreti” (pagina 14)
- ✓ Il foglio di lavoro: “la posta elettronica e i modem” (pagina 15)
- ✓ Il foglio di lavoro: “contare oltre il numero 31” (pagina 16)
- ✓ Il foglio di lavoro: “ancora di più sui numeri binari” (pagina 18)

I numeri binari

Introduzione

Prima di distribuire il foglio di lavoro di pagina 9 è utile che mostriate le regole del gioco a tutta la classe. Per questa attività, avete necessità di cinque carte, come mostrato qui sotto, con punti su un lato e niente sull'altro. Scegliete cinque studenti che si pongano uno a fianco all'altro di fronte al resto della classe. Date a ognuno di loro una delle carte in modo che appaiano alla classe nell'ordine seguente:



Discussione

Quale regola unisce il numero dei punti presente sulle carte? (Ogni carta ha il doppio dei punti della carta immediatamente alla destra).

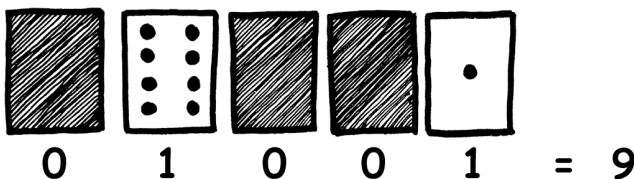
Quanti punti avrebbe la prossima carta se ne aggiungessimo una a sinistra? (32) e quella successiva? (64)...

Potete usare queste carte per scrivere numeri tenendone alcune con i punti rivolti verso la classe e girando le rimanenti dal lato del dorso. La somma dei punti visibili è il numero. Chiedete agli studenti di scrivere 6 (la carta 4 e la carta 2) poi 15 (la 8, la 4, la 2 e la 1), quindi 21 (16, 4 e 1)... La sola regola è che ogni carta deve essere o completamente visibile o completamente girata in modo che nessuno dei suoi punti sia visibile. Qual è il minimo numero di punti che si può ottenere? (Gli studenti forse risponderanno uno, ma la risposta esatta è zero).

Ora provate a contare partendo da 0.

Il resto della classe deve osservare attentamente come le carte cambino per capire se ci sia una regola alla base del numero di volte in cui ogni carta venga girata (ogni carta viene girata la metà delle volte di quella alla sua destra). Magari provate questo esercizio con più di un gruppo di studenti.

Una carta visibile, cioè esposta dal lato con i punti, si rappresenta con un uno. Una carta girata dal lato senza punti si rappresenta con uno zero. Questo è il sistema di numerazione binario.



Chiedete agli studenti di calcolare quale numero sia 01001 in decimale (9) e quale numero binario corrisponda al decimale 17 (10001)

Fate ancora altri esercizi per essere certi che tutti abbiano capito. Ci sono cinque attività opzionali per l'approfondimento. Gli studenti dovrebbero svolgerne più che possono.

Foglio di lavoro: i numeri binari

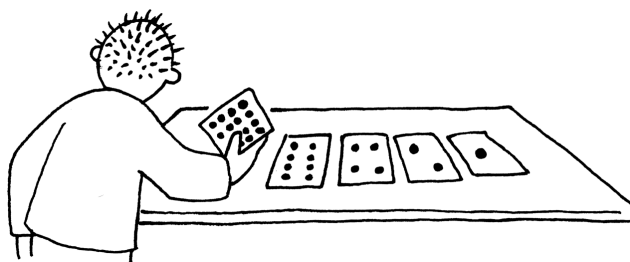
Imparare a contare

Pensavate di saper contare? Bene, ora imparerete un nuovo modo per farlo!

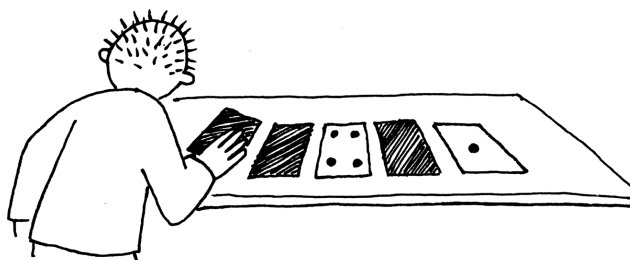
Sapevate che i computer usano solo zero e uno? Tutto quello che vedete o sentite dal vostro computer, parole, immagini, numeri, filmati e anche suoni, sono immagazzinati usando solo questi due numeri! Queste attività vi spiegheranno come mandare messaggi segreti ai vostri amici usando lo stesso metodo dei computer.

Istruzioni

Ritagliate le carte dal vostro foglio o ponetele davanti a voi in modo che la carta con 16 punti sia alla vostra sinistra come vedete qui:



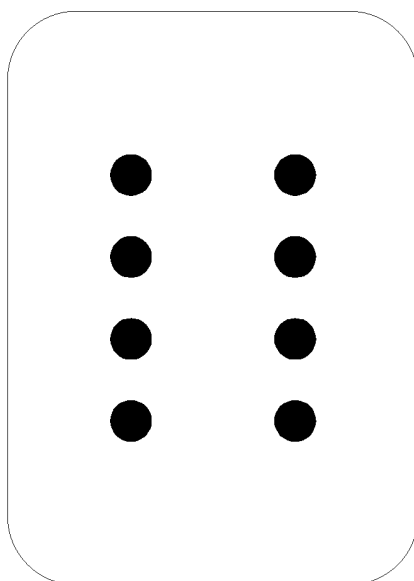
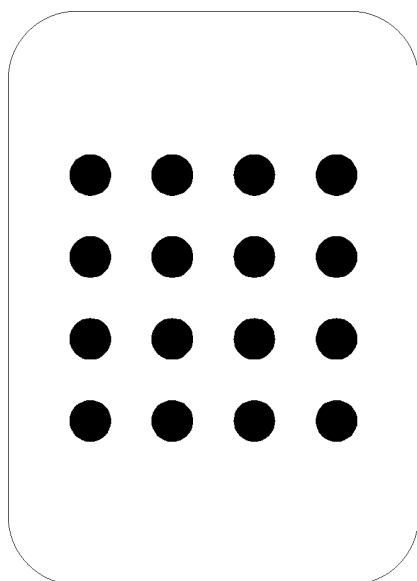
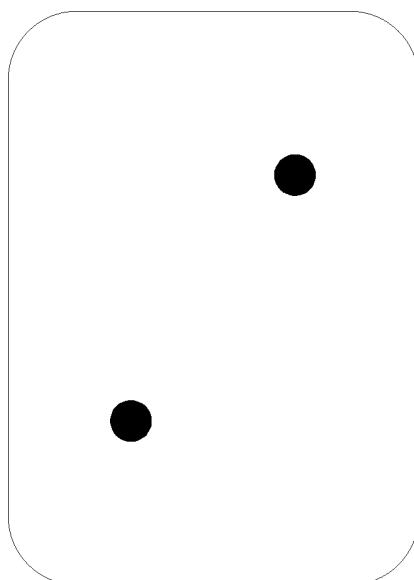
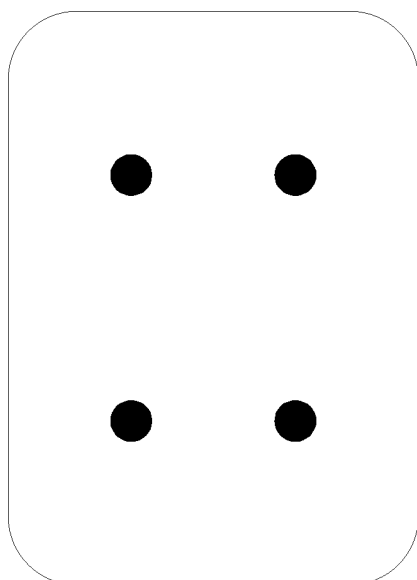
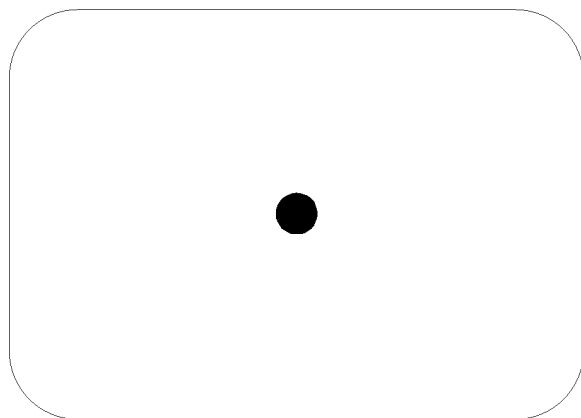
Assicuratevi che le carte siano esattamente poste nell'ordine indicato nella figura. Ora, mantenendo le carte nello stesso ordine, rovesciate alcune carte sul dorso in modo che siano visibili esattamente 5 punti.



Nello stesso modo provate ora a far comparire 3 punti, poi 12 e 19. Quanti modi possibili esistono per far comparire ogni numero? Qual è il massimo numero di punti che riuscite a far comparire? Qual è il minimo? C'è qualche numero di punti che non potete ottenere fra il numero minimo e quello massimo?

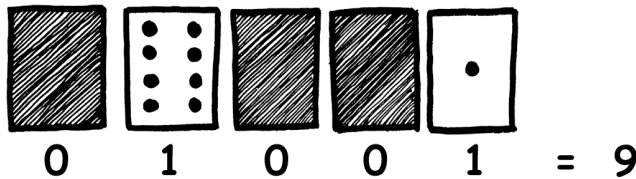
Extra per esperti: Provate con i numeri 1, 2, 3, 4 in ordine. Potete descrivere un metodo per aggiungere un punto alla volta?

Pagina da fotocopiare: numeri binari



Foglio di lavoro: lavorare con i numeri binari

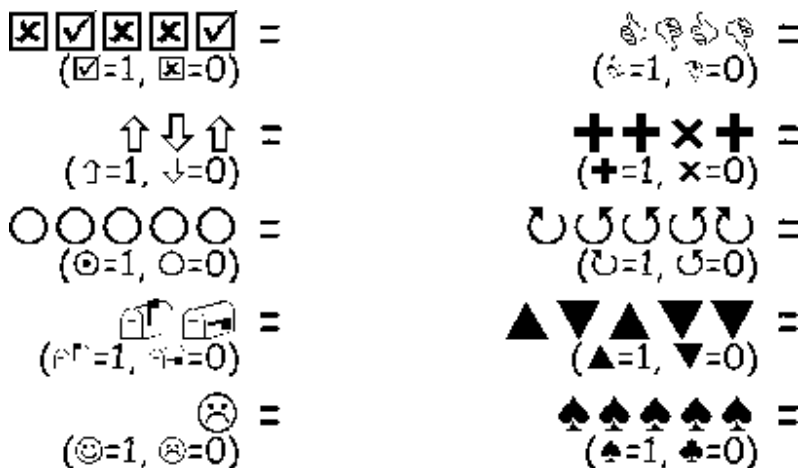
Il sistema binario usa i numeri zero e uno per rappresentare quali carte sono dal lato con i punti e quali sul dorso. 0 corrisponde a una carta sul dorso e 1 significa che potete vedere i punti. Per esempio:



Potete scoprire cosa è **10101**? E **11111** a quale numero corrisponde?

In quale giorno e in quale mese siete nati? Scrivetelo in binario. Scoprite come si scrive in binario il giorno del compleanno dei vostri amici.

Provate ora a scoprire questi numeri in codice:



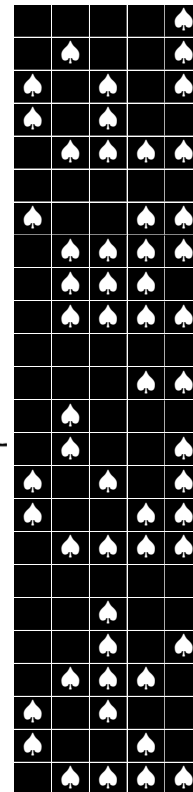
Extra per esperti: Usando una serie di regoli di lunghezza 1, 2, 4, 8 e 16 unità mostrate come potete giustapporli per realizzare sequenze di ogni lunghezza fino a 31 unità. Oppure con una bilancia a due piatti e cinque pesi da 1, 2, 4, 8, 16 unità mostrate come si possono misurare pesi da 0 unità a 31 unità.

Foglio di lavoro: inviare messaggi segreti

Tom è intrappolato all'ultimo piano di un grande magazzino. Mancano pochi giorni a Natale e voleva tornare a casa con alcuni regali ma non si è accorto dell'orario di chiusura ed è rimasto all'interno. Ha provato a chiamare, a urlare ma non c'è nessuno. È ormai notte, Tom vede nel palazzo di fronte una ragazza specialista in informatica che sta lavorando. Come può attirare la sua attenzione? Tom si guarda attorno per vedere cosa possa usare. Ha una idea geniale! Può usare le luci dell'albero di Natale per mandarle un messaggio! Tom connette le luci in modo da poterle accendere e spegnere tutte insieme. Quindi usa un semplice codice binario che è sicuro che la ragazza dall'altro lato della strada è capace di comprendere. Volete provarci anche voi?



1	2	3	4	5	6	7	8	9	10	11	12	13
a	b	c	d	e	f	g	h	i	j	k	l	m
14	15	16	17	18	19	20	21	22	23	24	25	26
n	o	p	q	r	s	t	u	v	w	x	y	z



Foglio di lavoro: posta elettronica e modem

Anche i computer collegati a Internet attraverso un modem usano il sistema binario per spedire i messaggi. La sola differenza è che in questo caso si usano dei suoni. Un "bip" acuto viene usato per spedire un 1 mentre uno di tono basso per uno 0. Questi suoni sono molto veloci, così veloci che i vecchi modem attaccati alla presa del telefono durante la fase di collegamento facevano udire solamente un fastidioso gracchiare. Se non l'avete mai sentito, provate a chiamare un fax, le macchine del fax usano questo tipo di segnali per spedire informazioni.



Utilizzando lo stesso codice usato da Tom per chiedere aiuto nel grande magazzino, provate a mandare un messaggio di posta elettronica a un vostro amico o a una vostra amica. Non dovete essere veloci come i veri modem, altrimenti non riuscirete a interpretare il messaggio!



Foglio di lavoro: contare oltre il numero 31

Osservate ancora la sequenza delle carte dei numeri binari. Se doveste creare la prossima carta della sequenza quanti punti dovrebbe avere? E quale è la carta ancora successiva? Quale è la regola che seguite per creare le nuove carte? Come potete osservare poche carte sono necessarie per contare fino a numeri molto grandi.

Se osservate la sequenza, potrete osservare una interessante relazione:

1, 2, 4, 8, 16

Provate a sommare: $1 + 2 + 4 = ?$ Qual è il risultato?

E ora provate: $1 + 2 + 4 + 8 = ?$

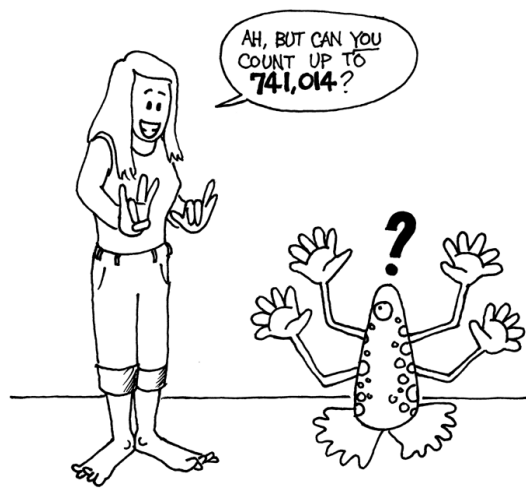
Cosa succede se sommate tutte le carte?

Ora proviamo a contare con le dita, non nel modo solito ma usando il binario. Voi siete normalmente capaci di contare fino a dieci? Bene con il binario potete contare fino a numeri molto più grandi di dieci, senza la necessità di essere degli alieni con tante dita! Usando i numeri binari potete contare da 0 a 31 usando una mano sola facendo in modo che ogni dito rappresenti una delle carte. Il pollice corrisponde alla carta con un punto, l'indice a quella con 2, il medio a quella con 4 e così via. Potete rappresentare con una mano 32 diversi numeri (non dimenticate che anche lo zero è un numero!)

Provate allora a contare usando le dita, il dito alzato rappresenta uno, cioè è come se la carta corrispondente fosse visibile, se il dito è abbassato è uno zero.

Usando entrambe le mani potete contare da 0 a 1023! Sono 1024 numeri!

Se aveste le dita dei piedi snodate (e qui dovrete veramente essere alieni) potreste ottenere numeri molto più alti. Se con una mano potete ottenere 32 numeri, con due mani $32 \times 32 = 1024$ numeri, fino a quale numero può contare la signorina "dita-dei-piedi-snodate" che conta in binario con tutte e 20 le dita delle mani e dei piedi?



(e tu sai contare fino 741.014?)

Foglio di lavoro: ancora di più sui numeri binari

1. Un'altra interessante proprietà dei numeri binari la potete osservare quando aggiungete a un numero uno zero dopo l'ultima cifra a destra. Se lavorate con i numeri in base 10 (i numeri decimali), quando aggiungete uno zero moltiplicate il numero per 10. Per esempio 9 diventa 90, 30 diventa 300.

Ma cosa succede quando aggiungete uno zero alla dell'ultima cifra di un numero binario? Provate con questo esempio:

$$\begin{array}{ccc} \mathbf{1001} & \rightarrow & \mathbf{10010} \\ (9) & & (?) \end{array}$$

Fate le vostre ipotesi. Quale è la regola? Perché secondo voi accade questo?

2. Ogni carta che abbiamo usato rappresenta un "bit" del computer (infatti bit è l'abbreviazione di "binary digit" che in inglese significa semplicemente cifra binaria). Il nostro codice per rappresentare l'alfabeto usa solo cinque carte cioè cinque bit. I computer usano più simboli, devono riconoscere le lettere minuscole da quelle maiuscole, quelle accentate, i numeri, i simboli della punteggiatura, i simboli speciali come \$ oppure %.

Osservate una tastiera e provate a calcolare quanti caratteri compongono l'alfabeto che un computer deve rappresentare. Quanti bit sono necessari per un computer per poter rappresentare tutti questi simboli?

La maggior parte dei computer oggi usa una rappresentazione standard chiamata ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange, tradotto suona come "codice americano standard per lo scambio di informazioni"), che è basato su questo numero di bit per ogni carattere. Questo codice, nato per la lingua inglese, va bene anche per l'italiano, il francese, il tedesco, lo spagnolo, ... ma in alcune lingue hanno molti più simboli e quindi hanno necessità di codici composti da più bit.

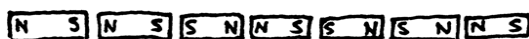


Cosa c'entra tutto questo?

I computer oggi usano il sistema binario per rappresentare l'informazione. È chiamato binario perché usa due sole cifre. Viene anche chiamato base due (gli umani di solito usano il sistema decimale cioè la base dieci). Ogni cifra (zero o uno) viene chiamata *bit* (**b**inary **d**igit, cioè cifra binaria). Un bit è rappresentato nella memoria principale di lavoro di un computer come un transistor che è acceso o spento o da un condensatore carico o non carico.



Quando i dati devono essere trasmessi su di una linea telefonica o attraverso un canale radio, vengono usati toni di diversa frequenza, acuti o bassi per rappresentare cifre uno o zero. Sui dischi magnetici (dischi rigidi) e sui nastri i bit sono rappresentati come direzioni del campo magnetico su una superficie ricoperta di materiale magnetizzabile. I due valori sono memorizzati con elementi magnetizzati in direzione nord-sud o sud-nord.



I CD musicali, i CD-ROM e i DVD registrano i bit in modo ottico: le parti della superficie corrispondenti a un bit riflettono o no la luce.

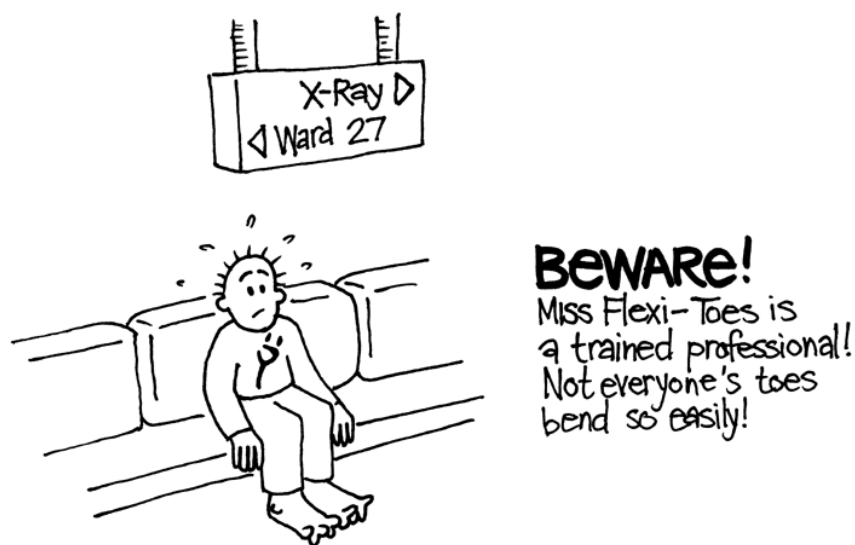


I computer usano solo due valori perché è molto più facile costruire apparecchi che funzionino così. Potremmo costruire dei CD che riflettano i raggi di luce con 10 intensità diverse per rappresentare i numeri da 0 a 9, ma gli apparati per leggerli o scriverli dovrebbero essere molto più precisi e costosi. L'altra cosa che potete aver notato è che sebbene noi diciamo che i computer memorizzano i dati usando solo numeri zero e uno, non ci sono nei computer numeri zero o uno ma solo tensioni (voltaggi) alti e bassi, magnetizzazioni di tipo nord o sud e così via. Ma è più rapido scrivere "0" o "1" piuttosto che brillante/non brillante. Tutto per i computer è rappresentato usando questi bit: documenti, immagini, canzoni, video, numeri e anche i programmi e le applicazioni che usiamo non sono altro che sequenze di tanti numeri binari.

Un bit non può rappresentare molto, ma i bit sono normalmente raggruppati otto a otto. Ogni gruppo di otto bit, denominato byte, può rappresentare numeri da 0 a 255.

La velocità di un computer dipende anche da quanti bit può elaborare contemporaneamente. Per esempio, molti dei computer oggi si dicono "a 32 bit" perché elaborano a ogni passo numeri di ampiezza fino a 32 bit. I computer a 16 bit, quando devono elaborare numeri a 32 bit, devono spezzare i dati e fare più operazioni e quindi sono più lenti.

I bit e i byte sono tutto ciò che il computer usa per memorizzare e trasmettere numeri, testo e tutte le altre informazioni. In alcune attività, più avanti nel testo, vi mostreremo come diversi tipi di informazione possano essere rappresentati dai computer.



(sala di attesa per i raggi X: "Attenzione, la signorina dita-dei-piedi-snodate è una contorsionista professionista. Non tutti possono piegare le dita dei piedi così facilmente!)

Soluzioni e suggerimenti

Numeri binari (pagina 9)

3 si rappresenta con le carte 2 e 1

12 si rappresenta con le carte 8 e 4

19 si rappresenta con le carte 16, 2 e 1

C'è una sola rappresentazione per ogni numero.

Il massimo numero rappresentabile è 31, il più piccolo è 0. È possibile rappresentare tutti i numeri naturali compresi fra questi estremi e tutti hanno una rappresentazione unica.

Extra per esperti: Per incrementare il numero di 1 occorre girare ogni carta a partire da destra fino a quando una carta viene girata dal dorso al recto (se la carta che mostrava i punti viene coperta si continua, se invece una carta che era coperta mostra ora i punti il procedimento termina).

Lavorare con i numeri binari (pagina 13)

10101 = 21, 11111 = 31

Inviare messaggi segreti (pagina 14)

Ecco il messaggio spedito da Tom: AIUTO SONO CHIUSO DENTRO

Contare oltre il numero 31 (pagina 16)

Se sommate tutti i numeri sulle carte la somma sarà sempre di una unità minore del prossimo numero nella sequenza.

La signorina dita-dei-piedi-snodate può rappresentare $1024 \cdot 1024 = 1,048,576$ numeri e può contare da 0 a 1,048,575!

Ancora di più sui numeri binari (pagina 18)

Quando viene aggiunto uno zero il numero rappresentato viene raddoppiato.

Tutti gli 1 del numero (i punti visibili delle carte) assumono una posizione che rappresenta un valore doppio (occorre cambiare ogni carta con quella alla sinistra), il valore totale viene quindi duplicato (in base 10 aggiungere uno zero alla destra equivale a moltiplicare per 10, la base).

Un computer ha necessità di 7 bit per registrare tutti i caratteri. Questo consente fino a 128 caratteri diversi. Di solito si usano 8 bit con un bit inutilizzato.

Attività 2

Colorare con i numeri – *La rappresentazione delle immagini*

Sommario

I computer memorizzano i disegni, le fotografie e le altre figure usando solo numeri. Questa attività mostra come fanno.

Competenze richieste:

Gli studenti devono essere in grado di:

- ✓ Contare
- ✓ Disegnare

Età

- ✓ A partire da 7 anni

Materiale

- ✓ Immagini da proiettore ottenute fotocopiando il prototipo “colorare con i numeri” (pagina 27)

Ogni studente deve avere:

- ✓ Il foglio di lavoro “Kid Fax” (pagina 28)
- ✓ Il foglio di lavoro: “E ora fate i vostri disegni” (pagina 29)

Colorare con i numeri

Introduzione

Domande di discussione:

1. Come funziona una macchina fax?
2. In quali situazioni i computer memorizzano immagini? (Programmi per il disegno, videogiochi, sistemi multimediali)
3. Come fanno i computer a memorizzare le immagini visto che possono solo usare numeri?

(Si può anche mostrare agli studenti come si spediscono e si ricevono i fax o fare in modo che gli studenti stessi inviino e ricevano fax come preparazione per questa attività)

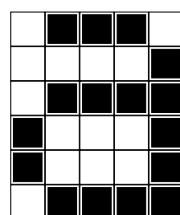
Esempio pratico con l'uso di un videoproiettore.



Gli schermi di un computer sono divisi in una griglia di tanti punti chiamati *pixel* (**picture element**, elementi di immagine).

In uno schermo in bianco e nero ogni punto può essere o bianco o nero.

La lettera "a" è stata ingrandita qui sopra per mostrare i pixel che la compongono. Quando un computer memorizza una immagine tutto ciò che deve immagazzinare è l'informazione di quali punti debbano essere bianchi e quali neri.

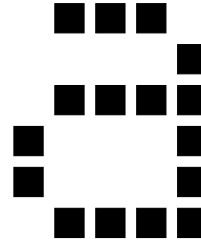
	1, 3, 1 4, 1 1, 4 0, 1, 3, 1 0, 1, 3, 1 1, 4
---	---

La figura qui sopra mostra come una immagine possa essere rappresentata tramite numeri. La prima riga è composta da un pixel bianco, tre neri e uno bianco. Viene quindi rappresentata come 1, 3, 1.

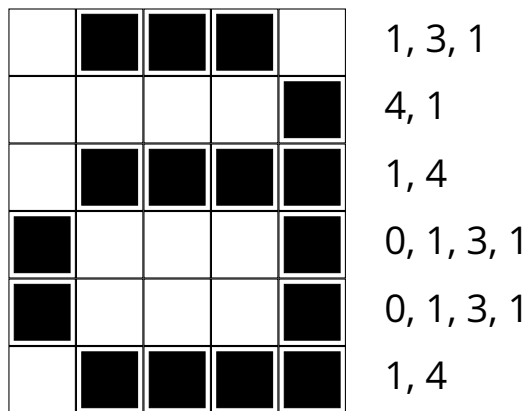
Il primo numero è sempre relativo al numero dei pixel bianchi all'inizio della linea. Se il primo pixel è nero la linea inizierà con uno zero.

Il foglio di lavoro di pagina 28 contiene alcune immagini che gli studenti possono decodificare usando il metodo appena mostrato.

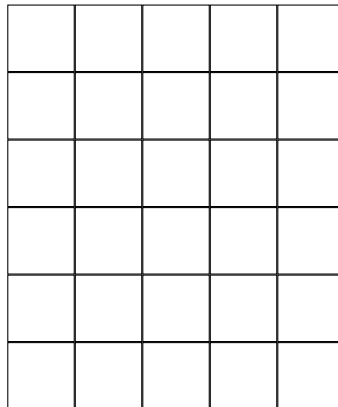
Colorare con i numeri



È la lettera "a" come appare sullo schermo di un computer e ingrandita per mostrare i pixel che la compongono



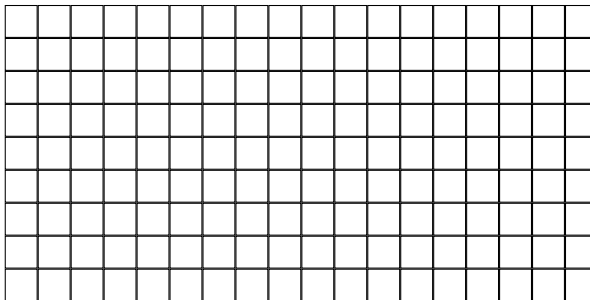
Questa è la stessa immagine codificata mediante numeri



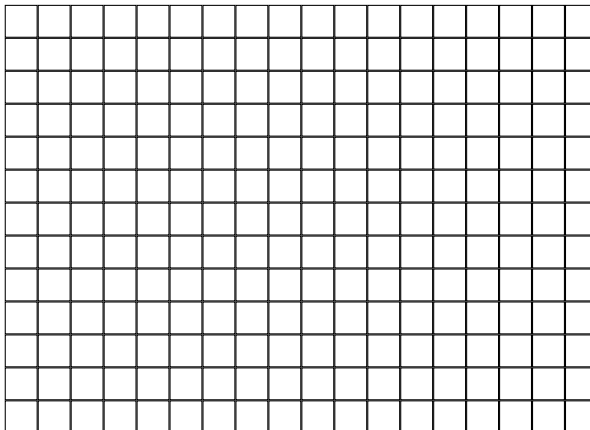
(Griglia vuota per scopi didattici)

Foglio di lavoro: Kid Fax

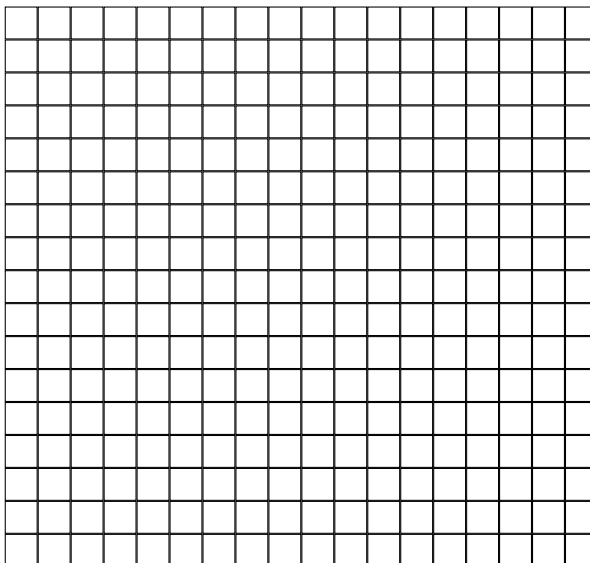
La prima figura è la più facile e l'ultima è la più complessa. È facile fare errori e quindi vi consigliamo di usare una matita colorata e di avere una gomma a portata di mano!



4, 11
 4, 9, 2, 1
 4, 9, 2, 1
 4, 11
 4, 9
 4, 9
 5, 7
 0, 17
 1, 15



6, 5, 2, 3
 4, 2, 5, 2, 3, 1
 3, 1, 9, 1, 2, 1
 3, 1, 9, 1, 1, 1
 2, 1, 11, 1
 2, 1, 10, 2
 2, 1, 9, 1, 1, 1
 2, 1, 8, 1, 2, 1
 2, 1, 7, 1, 3, 1
 1, 1, 1, 1, 4, 2, 3, 1
 0, 1, 2, 1, 2, 2, 5, 1
 0, 1, 3, 2, 5, 2
 1, 3, 2, 5



6, 2, 2, 2
 5, 1, 2, 2, 2, 1
 6, 6
 4, 2, 6, 2
 3, 1, 10, 1
 2, 1, 12, 1
 2, 1, 3, 1, 4, 1, 3, 1
 1, 2, 12, 2
 0, 1, 16, 1
 0, 1, 6, 1, 2, 1, 6, 1
 0, 1, 7, 2, 7, 1
 1, 1, 14, 1
 2, 1, 12, 1
 2, 1, 5, 2, 5, 1
 3, 1, 10, 1
 4, 2, 6, 2
 6, 6

Variazioni e estensioni.

1. Provate a far colorare i ragazzi su un foglio di carta da lucido (o qualsiasi altro supporto trasparente) posto sulla griglia, così l'immagine finale non avrà la griglia e sarà quindi più chiara.
2. Invece di colorare gli elementi della griglia su di un foglio i ragazzi possono attaccare quadrati di carta adesiva o porre oggetti colorati su una griglia più grande.

Punti di discussione.

C'è di solito un limite alla lunghezza massima della sequenza di pixel che è stesso colore data dal massimo numero binario rappresentabile. Come rappresentereste una sequenza di dodici pixel neri sapendo che il vostro sistema usa solo tre bit per ogni sequenza e quindi il numero massimo è sette? (Un buon metodo è quello di scrivere la sequenza come una sequenza di sette pixel neri, zero pixel bianchi e infine cinque neri).

Cosa c'entra tutto questo?

Una macchina per il fax è un semplice computer che scandisce una pagina in bianco e nero e la trasforma in una griglia di circa 1000x2000 pixel, trasforma le righe in numeri, come abbiamo visto, quindi trasferisce i numeri a una macchina simile usando un modem che ritrasforma i numeri in pixel e quindi stampa i pixel su un foglio. Spesso i fogli inviati contengono lunghe sequenze di pixel bianchi (per esempio i margini) o di pixel neri (una linea orizzontale).

Anche le immagini a colori hanno molti pixel ripetuti. Per risparmiare sullo spazio di memorizzazione necessario per mantenere queste immagini i programmatori hanno sviluppato una serie di altri metodi di compressione. Il metodo descritto in questa attività è conosciuto in letteratura come "run-length encoding" (codifica di lunghezza delle sequenze) ed è un metodo molto efficace per comprimere le immagini. Se non comprimessimo le immagini sarebbe stato necessario molto più tempo per trasmettere le immagini e molto più spazio per memorizzarle. Le immagini non compresse sarebbero poco pratiche per i fax o per inserire fotografie in una pagina web. Per esempio tipicamente i fax trasmettono un settimo dei dati che sarebbero necessari per trasmettere le immagini pixel per pixel. Senza la compressione occorrerebbe un tempo sette volte maggiore per trasmettere o ricevere un fax.

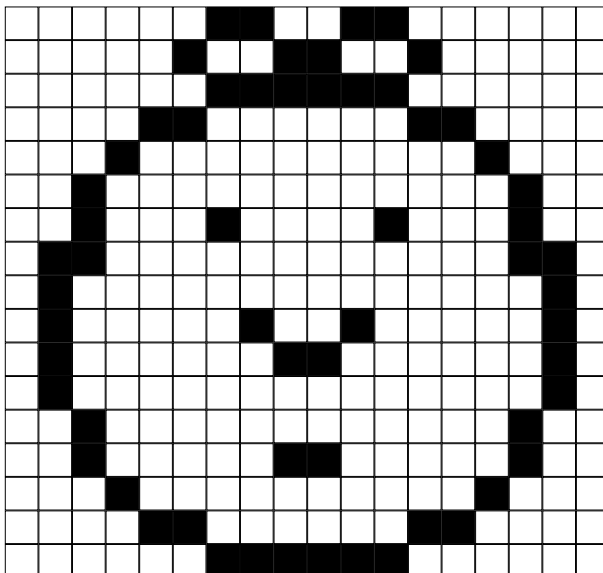
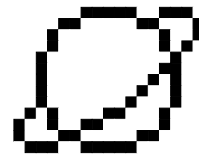
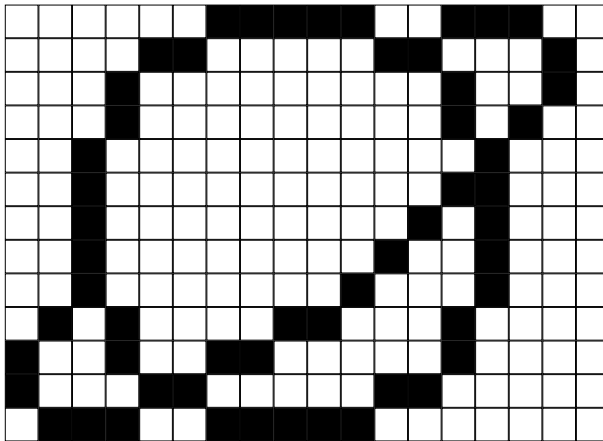
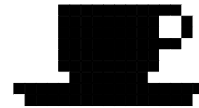
Le fotografie e le immagini sono di solito compresse con tecniche che consentono di usare da un decimo a un centesimo dei dati dell'immagine originale (usando altre tecniche). Questo consente a molte immagini di poter venir memorizzate su di un disco e anche di impiegare una quantità molto minore di tempo per poterle recuperare e vedere.

Un programmatore può scegliere fra varie tecniche di compressione per trovare quella più adatta alle immagini da trasmettere o da memorizzare.



Soluzioni e suggerimenti

Risposte al Foglio di lavoro: Kid Fax



Attività 3

Puoi dirlo nuovamente! — *La compressione del testo*

Sommario

Dato che i computer hanno uno spazio limitato per poter immagazzinare le informazioni, devono rappresentarle nel modo più efficiente possibile. Si usano quindi metodi di compressione dell'informazione. Comprimendo i dati per memorizzarli e decomprimendoli per rileggerli, si possono conservare più informazioni o spedirle più velocemente attraverso Internet.

Competenze richieste

- ✓ Copiare testo scritto

Età

- ✓ A partire da 9 anni

Materiale

- ✓ Immagini da proiettore ottenute fotocopiando il prototipo “puoi dirlo nuovamente!” (pagina 37)

Ogni studente deve avere:

- ✓ Il foglio di lavoro: puoi dirlo nuovamente! (pagina 38)
- ✓ Il foglio di lavoro: extra per esperti (pagina 40)
- ✓ Il foglio di lavoro: Apelle, figlio di Apollo (pagina 41)
- ✓ Il foglio di lavoro: extra per veri esperti (pagina 42)

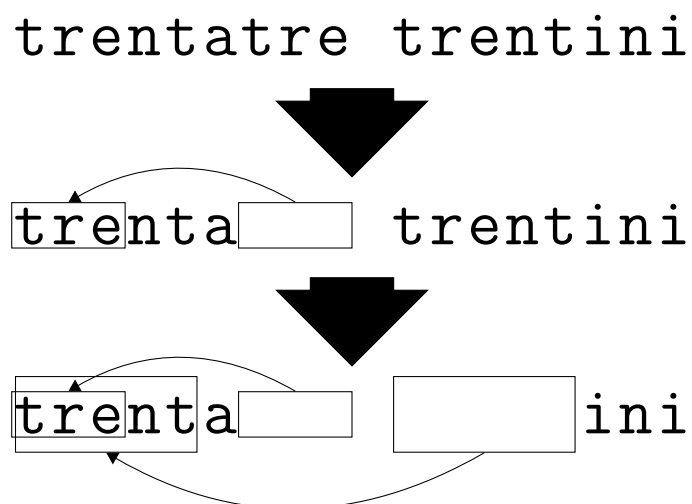
Puoi dirlo nuovamente!

Introduzione

I computer devono memorizzare e trasmettere tantissimi dati. Quindi per ottimizzare il lavoro, devono evitare di usare troppo spazio o impiegare troppo tempo per trasmetterli lungo le reti. Questa attività mostra come si possono comprimere i testi.

Dimostrazione e discussione.

Mostrate lo scioglilingua di pagina 37. Osservate quali sequenze di lettere sono ripetute nel testo. Potete trovare i gruppi di due o più lettere sono ripetute o anche intere parole o gruppi di parole. Sostituite le sequenze ripetute tracciando loro un rettangolo attorno come nel diagramma che segue.



**Immagine da proiettare (o fotocopiare):
puoi dirlo nuovamente!**

trentatre trentini

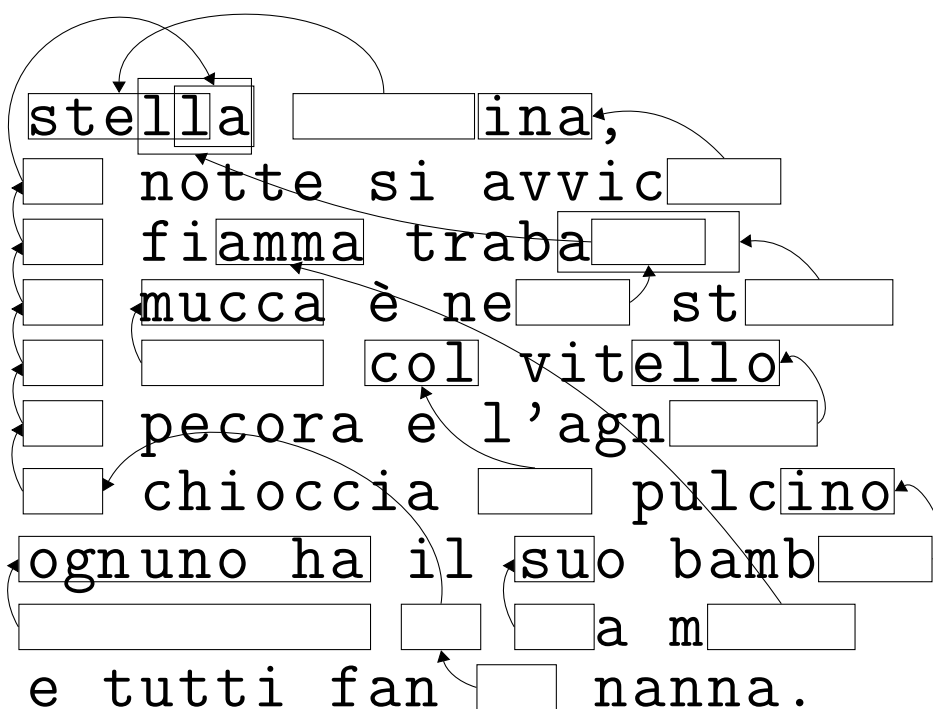
entrarono a Trento

tutti e trentatre

trotterellando

Foglio di Lavoro: puoi dirlo nuovamen- te!

Molte delle lettere e delle parole della seguente poesia sono mancanti. Potete inserire le lettere e le parole mancanti per completare correttamente il testo? Troverete le lettere e le parole mancanti nel riquadro puntato dalla freccia:



Ora potete scegliere altre semplici poesie o ninna nanne e creare un puzzle simile. Controllate che le frecce puntino sempre ad una parte precedente del testo. La vostra poesia deve sempre essere ricomposta scorrendo il testo da sinistra a destra, dall'alto in basso, nell'ordine solito della lettura.

Sfida: Provate a vedere quante poche parole e lettere della poesia originale occorre tenere!

Altri suggerimenti: usate degli scioglilingua (sopra la panca..., se il coniglio gli agli piglia..., ...) oppure poesie o parole di canzoni con ripetizioni.

Suggerimento: Tentate di evitare un sovraffollamento di frecce. Lasciate molto spazio fra le lettere e fra le parole così che ci sia lo spazio sufficiente per disegnare i riquadri e le frecce che puntano ad esse.

È più facile disegnare il puzzle se prima scrivete la poesia e poi decidete quali debbano essere i riquadri da tracciare.

Foglio di Lavoro: extra per esperti

Come risolvereste questo puzzle?

Talvolta le parti mancanti del testo puntano a una porzione di se stesse. In questo caso occorre copiare lettera per lettera da sinistra a destra per poter ritrovare il testo originario. In questo modo ogni lettera è già disponibile quando è necessaria per proseguire. Questo metodo è utile quando i computer si trovano lunghe sequenze di un particolare carattere o di un gruppo di caratteri.

Potete provare a creare altri esempi per conto vostro.

I computer rappresentano i riquadri e le frecce con numeri, per esempio:

Banana

può essere scritto come **Ban(2,3)**. "2" significa che occorre tornare indietro di due caratteri per iniziare la copia.

Ban___

e "3" significa che occorre copiare tre caratteri consecutivi.

Bana__

Banan_

Banana



(cosa stai facendo?)

Sto comprimendo le banane)

Siccome vengono utilizzati due numeri per indicare che ci sono caratteri ripetuti, solo gruppi di due o più lettere vengono compressi, altrimenti non c'è alcun risparmio di spazio. Infatti la lunghezza di un testo si allungherebbe se utilizzassimo due numeri al posto di una lettera.

Scrivete alcune parole nel modo usato da un computer per comprimerle, cercate le parole più interessanti per la compressione. Riescono i vostri amici a ricostruire le parole che avete compresso?

Foglio di lavoro: Apelle, figlio di Apollo

Quante parole sono veramente necessarie qui?

Pensate di essere un computer che voglia mettere sul proprio disco più informazioni possibili. Scorrete il testo e trovate tutte le sequenze di due o più lettere che sono già comparse nel testo. Queste sequenze non sono più utili e possono essere sostituite con un puntatore (un riquadro vuoto e una freccia che indica un altro riquadro con le lettere da copiare). Il vostro scopo è di trovare più sequenze che potete.

Apelle figlio di Apollo

Fece una palla di pelle di pollo

Tutti i pesci vennero a galla

Per vedere la palla di pelle di pollo

Fatta da Apelle figlio di Apollo

Foglio di lavoro: extra per veri esperti

Siete pronti per una compressione veramente dura?

La favola che segue è stata analizzata da un programma per computer (uno di quelli veri con la spina) che ha trovato che ci sono almeno 499 sequenze e 2281 lettere che possono essere eliminate perché appartengono a sequenze ripetute (la divisione fra righe e paragrafi viene considerata come uno spazio). Quante sequenze riuscite a trovare? Quanti caratteri in totale potete eliminare? Ricordatevi che solo le sequenze di due o più caratteri ripetuti possono essere eliminate. Buona Fortuna!

C'erano una volta, tanto tempo fa, tre porcellini che volevano andare a vivere da soli. Il primo porcellino non era molto furbo e decise di costruire la sua casa con la paglia perché costava poco. Anche il secondo porcellino non era furbo e costruì la sua casa di legno perché aveva un look ecologico ed era di moda. Il terzo porcellino era più furbo dei suoi due fratelli e comperò un carico di mattoni in una città vicina coi quali costruì una casa modesta ma confortevole.

Poco tempo dopo la festa di inaugurazione della casa, il primo porcellino era assorto su una sedia leggendo un libro quando udì bussare alla porta. Era il lupo cattivo, naturalmente.

"Porcellino, porcellino, lasciami entrare!" gridò il lupo.

"Neanche per un pelo della mia cotenna!" rispose il primo porcellino.

"Allora sbufferò, soffierò e abatterò la tua casa" ruggì il lupo e sbuffò, soffiò e la casa in breve crollò. Il primo porcellino corse più forte che poté verso la casa di legno e presto fu al sicuro lì col fratello. Ma non passò molto che il lupo arrivò anche lì.

"Porcellino, porcellino, lasciami entrare!" gridò il lupo.

"Neanche per un pelo della mia cotenna!" rispose il secondo porcellino. "Allora sbufferò, soffierò e abatterò la tua casa" ruggì il lupo e sbuffò, soffiò e presto della casa non rimase che un mucchio di legna per il camino. Il due porcellini terrorizzati corsero più forte che poterono verso la casa di mattoni, ma il lupo era sempre alle loro calcagna e appena i due porcellini arrivarono alla casa del loro fratello, anche il lupo arrivò ancora una volta sulla soglia.

"Porcellino, porcellino, lasciami entrare!" gridò il lupo.

"Neanche per un pelo della mia cotenna!" rispose il terzo porcellino.

"Allora sbufferò, soffierò e abatterò la tua casa" ruggì il lupo e sbuffò, soffiò e, naturalmente, la casa era fatta di mattoni e il lupo rimase senza fiato. Allora ebbe un'idea: il camino! Si arrampicò su una quercia e salì sul tetto. Ma il terzo porcellino che era molto sensibile ai temi ambientali invece del camino aveva installato pannelli solari. Frustrato il lupo scivolò e cadde dal tetto rompendosi la zampa sinistra, danneggiando irreparabilmente la propria autostima. Quando il lupo se ne andò i porcellini risero e pensarono a quanto sicuro sia vivere in città dove i lupi si possono trovare solo allo zoo. E così fecero e, naturalmente, vissero felici e contenti.

Cosa c'entra tutto questo?

La capacità di memorizzazione dei computer sta crescendo ad un ritmo incredibile negli ultimi 25 anni la quantità di dati immagazzinabili è cresciuta di circa un milione di volte, ma continuiamo a trovare informazioni da mettere nei computer. I computer possono contenere interi libri o intere biblioteche e ora musica e film a patto che abbiano spazio a sufficienza. I grandi archivi (file) rappresentano un problema sulla rete Internet perché hanno necessità di lunghi tempi per poter essere trasferiti. I computer inoltre stanno diventando sempre più piccoli: anche un telefono cellulare o un orologio da polso contengono moltissime informazioni.

C'è comunque una soluzione per risolvere questo problema. Invece di acquistare sempre più spazio per immagazzinare dati, per esempio dischi fissi o removibili, oppure invece di acquistare modem più veloci e comperare linee a velocità più elevata, possiamo comprimere i dati. Il processo di compressione e di decompressione dei dati è effettuato automaticamente dal computer. Tutto quello che possiamo notare è che il disco contiene più dati e che le pagine web vengono trasferite più velocemente. Quello che cambia è che il computer deve fare più calcoli.

Sono stati inventati molti metodi di compressione per i testi. Il metodo usato in questa attività, sfruttando il principio di riferirsi alla precedente occorrenza di sequenze ripetute, è noto con il nome di "codifica di Ziv-Lempel" o più semplicemente come codifica "ZL" perché è stata inventata da due professori israeliani (il prof. Ziv e il prof. Lempel), negli anni '70. È qualche volta indicata anche come "zip" nei personal computer e viene anche usata nelle immagini GIF e nei modem ad alta velocità. Nel caso dei modem, riduce la quantità di dati che devono essere scambiati sulla linea telefonica e quindi il collegamento risulta più veloce.

Alcuni altri metodi sono basati sull'idea che alcune lettere sono più frequenti di altre e quindi vengono associati codici più corti alle lettere più frequenti. Il codice Morse usa questo principio.

Soluzioni e suggerimenti

Puoi dirlo nuovamente! (pagina 38)

stella stellina,
la notte si avvicina
la fiamma traballa
la mucca è nella stalla
la mucca col vitello
la pecora e l'agnello
la chioccia col pulcino
ognuno ha il suo bambino
ognuno ha la sua mamma
e tutti fan la nanna.

Attività 4

La magia delle carte girate – *Il riconoscimento e la correzione degli errori*

Sommario

Quando i dati vengono memorizzati su un disco o trasmessi da un computer all'altro, noi assumiamo che non vengano modificati. Purtroppo talvolta le cose non vanno così e i dati vengono accidentalmente danneggiati. Questa attività mostra un trucco magico per poter scoprire quali dati sono stati rovinati e come poterli correggere.

Competenze richieste:

- ✓ Saper contare
- ✓ Riconoscere i numeri pari e i numeri dispari

Età

- ✓ A partire da 7 anni

Materiale

- ✓ 36 fermagli magnetici "da frigo" colorati su di un lato.
- ✓ Una superficie dove attaccare i magneti (di solito le lavagne bianche vanno bene).

Ogni coppia di studenti deve avere.

- ✓ 36 carte identiche colorate da un solo lato.

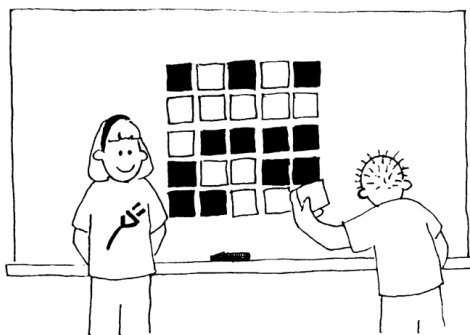
Il “trucco magico”

Dimostrazione

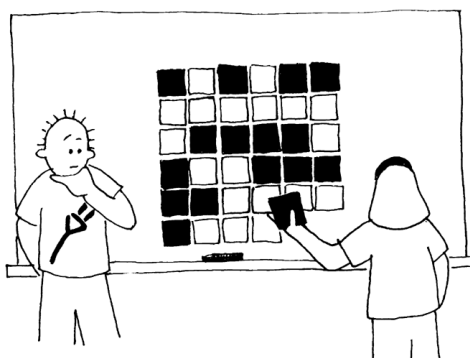
Questa è la vostra possibilità per essere un mago.

Avete bisogno di una pila di carte identiche colorate in modo diverso sui due lati (potete costruirle ritagliando un cartone colorato da un lato solo). Per mostrare il metodo agli studenti è più semplice usare calamite che abbiano colori diversi sui due lati esistono calamite “da frigo” con queste caratteristiche (perfette per questo uso).

1. Scegliete uno studente/una studentessa e chiedetegli/le di disporre le carte in modo che formino un quadrato 5×5 , scegliendo a caso i lati delle carte visibili.



Aggiungete voi una riga e una colonna, “giusto per fare le cose un po’ più difficili”.



Queste carte sono la chiave del trucco. Non le metterete a caso ma le disporrete in modo che tutte le righe e le colonne abbiano un numero pari di carte colorate.

2. Chiedete ora ad un altro studente o a un'altra studentessa di girare una sola delle carte mentre voi vi coprite gli occhi. La riga e la colonna dove è stata girata la carta contengono un numero dispari di carte colorate. In questo modo potete individuare la carta cambiata.

Ora gli studenti sono capaci di indovinare il trucco?

Insegnate il trucco agli studenti:

1. Lavorando in coppie gli studenti dispongono le carte a formare un quadrato 5x5.
2. Quante carte colorate ci sono in ogni riga e in ogni colonna? È un numero pari o dispari? Ricordate che zero è un numero pari.
3. Ora aggiungete una carta in ogni riga in modo che il numero di carte colorate di ogni riga sia pari. Questa carta aggiuntiva viene chiamata la "parità".
4. Aggiungete una riga in fondo in modo che ogni colonna contenga un numero pari di carte colorate.
5. Ora girate una carta. Come potete trovare la riga e la colonna della carta girata? (avranno un numero dispari di carte colorate). I bit di parità servono proprio a segnalare quando viene commesso un errore.
6. Ora fate a turni per provare il "trucco".

Proposte di estensione:

1. Provate a usare altri oggetti. Tutto ciò che ha due "stati" va bene. Per esempio si possono usare carte da gioco (rosso o nero), monete (testa o croce) o carte con 0 da un lato e 1 dall'altro (per correlare l'attività ai numeri binari).
2. Cosa accade se due o più carte vengono girate? (non è sempre possibile dire quali carte siano state cambiate. In generale si possono individuare due coppie di carte fra le quali ci sono quelle modificate. Cambiando quattro carte è possibile che i bit di parità siano corretti anche dopo la modifica e quindi l'errore rimarrebbe non riconosciuto).
3. Provate ora con una configurazione più ampia, per esempio 9×9 carte, che con la riga e la colonna aggiunte diventa 10×10 . (funziona per ogni dimensione, anche se la configurazione non è quadrata, cioè se ha un numero di righe e di colonne differenti).
4. Un altro esercizio interessante consiste nell'analizzare l'ultima carta in basso a destra. Per come è stato descritto il metodo questa carta viene calcolata per avere un numero pari di carte colorate nella colonna delle carte di parità di ogni riga. La stessa carta è anche la carta di parità per la ultima riga? (Sì sempre).
5. In questo esercizio abbiamo usato la parità (il numero di carte colorate in ogni riga e colonna deve essere pari). È se invece usassimo la disparità (il numero delle carte colorate nelle righe e nelle colonne deve essere dispari). Si può fare lo stesso ragionamento del punto precedente? L'ultima carta in basso a destra ricostruisce la disparità sia dell'ultima riga sia dell'ultima colonna?

(Può succedere, ma in generale funziona solo se il numero di righe e il numero di colonne sono entrambi pari o entrambi dispari. Funziona per il caso 5x5 ma anche per 5x9 o 4x6 ma non per 3x4).

Un esempio per esperti, tratto dalla vita reale.

La stessa tecnica è usata con i codici identificativi dei libri. Tutti i testi pubblicati hanno un codice numerico di identificazione formato da 10 cifre, di solito stampato nell'ultima di copertina. La decima cifra è una cifra di controllo, esattamente come il bit di parità nell'esercizio.

Questo significa che se ordinate un libro usando il codice ISBN (International Standard Book Number), l'editore può controllare che non abbiate commesso errori di trascrizione controllando la cifra di controllo. Questo fa in modo che l'errore possa essere scoperto prima che vi troviate con un libro diverso da quello che volevate ordinare.

Ecco come funziona il meccanismo di controllo:

Moltiplicate la prima cifra per dieci, la seconda per nove, la terza per 8 e così via fino alla nona cifra che va moltiplicata per due e sommate tutti questi prodotti insieme.

Per esempio per il codice ISBN 0-13-911991-4 si ottiene il seguente:

$$(0 \times 10) + (1 \times 9) + (3 \times 8) + (9 \times 7) + (1 \times 6) + \\ + (1 \times 5) + (9 \times 4) + (9 \times 3) + (1 \times 2) = 172$$

Dividete ora il risultato per 11. Quale è il resto di questa divisione?

$$172 \div 11 = 15 \text{ con il resto di } 7$$

Se il resto è zero, allora la cifra di controllo (detta *checksum* in inglese) è zero, altrimenti sottraete da 11 il resto per trovare il valore della cifra di controllo.

$$11 - 7 = 4$$

Ora potete controllare. È l'ultima cifra del codice ISBN? Sì! Allora funziona davvero!

Se l'ultima cifra fosse stata diversa da quattro, allora avremmo scoperto che il codice ISBN era errato, magari una cifra era stata copiata male.

È possibile che la cifra di controllo risulti 10 e quindi non basterebbe un solo numero per rappresentarla. In questo caso viene usato il carattere X.

Tutta la procedura incluso il numero 11 e la decisione di usare il carattere X quando il risultato è 10 è stata definita come parte dello

standard ISBN. Lo stesso metodo viene utilizzato quando il codice viene generato e quando viene controllato e consentendo così di evitare molti errori. Il numero 11 è un numero primo, quindi non ha fattori in comune con tutti i numeri usati nelle moltiplicazioni. Questo garantisce una migliore *distribuzione* dei valori del codice di controllo.

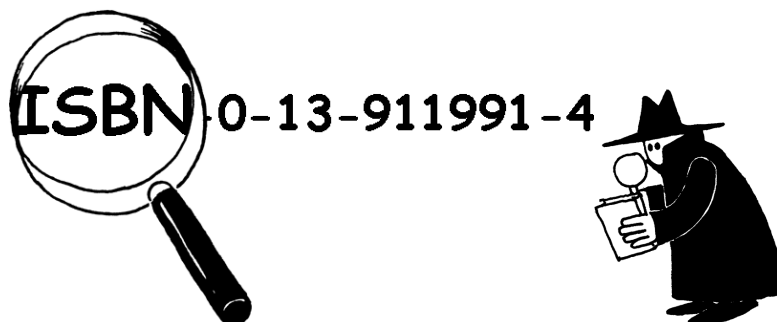


Un esempio di codice a barre di un prodotto alimentare

Un altro caso nel quale viene usata una cifra di controllo sono i codici a barre che trovate sui prodotti che comperate nei negozi e nei supermercati. Il concetto è lo stesso ma in questo caso si usa una diversa formula. Nei supermercati le casse hanno uno scanner: uno strumento che proiettando dei raggi laser, di solito di colore rosso, legge il codice a barre per riconoscere il prodotto acquistato e trovare il relativo prezzo. Se alla cassa lo scanner non legge bene il codice, la cifra di controllo non risulterà corretta e la cassa non farà il caratteristico bip. L'operatore tenterà di passare l'articolo nuovamente sotto lo scanner e alla fine se lo scanner non riesce a leggere il codice scriverà i numeri su di una tastiera, codice di controllo compreso. Così si controlla anche che l'operatore abbia scritto i numeri corretti!

Controllate quel libro

Agenzia Investigativa
Controllalibri



Cerchiamo e controlliamo i codici ISBN. Prezzi Modici.

Unitevi alla nostra agenzia, cercate nei libri in classe
o nella biblioteca i codici ISBN.

Le cifre di controllo sono corrette?

Talvolta vengono commessi errori nella trascrizione dei codici ISBN.

Alcuni degli errori comuni sono:

- ✗ copiare male una cifra;
- ✗ scambiare fra loro due cifre;
- ✗ aggiungere una cifra al numero per esempio copiandola due volte;
- ✗ non copiare una cifra del numero.

Riuscite a trovare libri che hanno la lettera X al posto della cifra di controllo? Non dovrebbe essere troppo difficile, in media un libro ogni 11 ha la lettera X.

Quale tipo di errori possono capitare senza che vengano riconosciuti dalla cifra di controllo? È possibile cambiare una sola cifra ed avere lo stesso la cifra di controllo corretta? Cosa succede se si scambiano fra loro due cifre (un errore molto comune)?

Cosa c'entra tutto questo?

Immaginate di depositare 10 euro nel vostro conto in banca. L'impiegato scrive l'importo del deposito e il dato viene spedito al computer centrale. Ma supponete ora che ci sia una interferenza lungo la linea e il codice che rappresenta 10 euro venga cambiato in 1000 euro. A voi magari piacerebbe, ma per la banca sarebbe chiaramente un problema.

È importante poter riconoscere gli errori che possono accadere nella trasmissione dei dati. Quindi occorre che il computer ricevente possa controllare che i dati che arrivano non siano stati rovinati da interferenze lungo la linea. Talvolta in caso di errore i dati possono essere ricevuti di nuovo, come nel caso dei dati scambiati tra due computer, il computer mittente può ritrasmetterli. Ci sono situazioni nelle quali non è possibile avere questa seconda possibilità, per esempio quando un disco o un nastro viene rovinato dall'esposizione a campi elettrici o magnetici, al calore o proprio fisicamente rotto. Talvolta anche se i dati possono teoricamente essere ritrasmessi non è conveniente farlo perché ci vorrebbe troppo tempo: ad esempio, i dati trasmessi da una sonda spaziale dal pianeta Giove hanno bisogno di almeno mezz'ora per arrivare fino a noi, sarebbe veramente noioso dover aspettare la ritrasmissione!

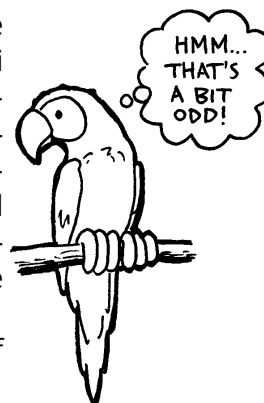
Occorrono quindi metodi per riconoscere quando un dato è rovinato (riconoscimento dell'errore o *error detection*) e per ricostruire il dato esatto da quello rovinato (correzione dell'errore o *error recovery*).

Lo stesso metodo usato nel gioco delle "carte girate" viene usato dai computer. Mettendo i bit in righe e colonne immaginarie, non solo possiamo vedere quando accadono errori ma anche dove sono. È così possibile ripristinare il valore corretto e quindi questo è un metodo di correzione dell'errore.

I computer spesso usano anche metodi di controllo più complessi che sono capaci di riconoscere e correggere più errori e non uno solo. I dischi rigidi di un computer hanno vaste aree dedicate alla correzione così che i dischi possano continuare a funzionare anche se parti del disco venissero danneggiate. Questi metodi sono comunque correlati al metodo di controllo della parità che abbiamo studiato. Siete bravi con l'inglese? Se sì apprezzerete questo gioco.

Q: What do you call this: "Pieces of nine, pieces of nine"?

A: A parrot error.



Soluzioni e suggerimenti

Ci sono errori che potrebbero essere non riconosciuti, accadono per esempio quando una cifra viene diminuita ed un'altra incrementata. La somma potrebbe essere uguale: per esempio, se aggiungete due alla ottava cifra (che viene moltiplicata per tre) e togliete tre alla nona cifra (moltiplicata per due) la somma non cambia.

Ecco la spiegazione del dialogo del pappagallo, se qualche studente (non madrelingua americano/a o inglese) ce l'ha fatta da solo/a a capirne il significato, probabilmente non avrà bisogno di studiare inglese fino all'università!

Traduzione letterale:

Domanda: come chiameresti questo: "pezzi da nove, pezzi da nove"?

Risposta: un errore da pappagallo.

Il pappagallo dice "è un po' strano".

Così sembra non avere senso!

Spiegazione:

Il "pezzo da otto" è un'antica moneta spagnola in uso anche in Messico. Nel libro "l'isola del tesoro" c'è un pappagallo ammaestrato per gridare a tutti quelli che entrano nella stanza "pezzi da otto, pezzi da otto". Nel film "I pirati dei Caraibi" i pirati devono trovare "nove pezzi da otto" e "pezzi da otto" è un'attrazione di Disneyland California.

La risposta si può leggere in due modi:

- ✓ un errore del pappagallo dell'isola del tesoro perché la moneta "pezzo da nove" non esiste;
- ✓ un errore di parità perché, come in una sorta di gioco di parole, "parroty error" suona come "parity error", cioè errore di parità.

Per quanto riguarda la frase del pappagallo, le parole "bit" e "odd" hanno entrambe due significati, uno nell'uso comune e uno proprio dell'informatica. La frase così si può leggere sia "è un po' strano" ma anche "è un bit dispari"!

Attività 5

Indovina indovinello — *La Teoria dell'Informazione*

Sommario

Quanta informazione c'è in un libro di 1000 pagine? C'è più informazione in un elenco telefonico di 1000 pagine oppure in una risma di carta di 1000 pagine bianche oppure nel libro di Tolkien *Il Signore degli Anelli*? Se riuscissimo a misurarla, allora potremmo stimare quanto spazio occorre per memorizzarne una data quantità. Per esempio, riuscite a leggere quello che segue?

L frs h l vcl mncnt

Probabilmente sì, perché non c'è "troppa informazione" nelle vocali. Questa attività presenta un modo per misurare la quantità di informazione contenuta.

Competenze richieste:

Gli studenti devono essere in grado di:

- ✓ Confrontare numeri e fare operazioni con intervalli numerici
- ✓ Trarre deduzioni logiche
- ✓ Fare domande

Età

- ✓ A partire da 10 anni

Materiale

- ✓ Nessun materiale è richiesto per la prima attività

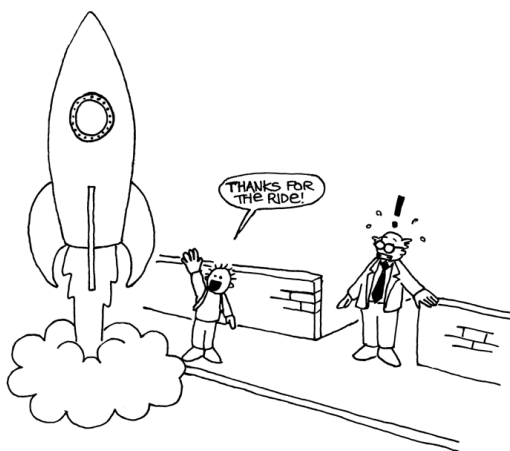
Per l'attività estesa, ciascun bambino ha bisogno di:

- ✓ Foglio di lavoro: alberi di decisione (pagina 63)

Indovina indovinello

Discussione

1. Discutete con i bambini cosa pensano che sia l'informazione.
2. Come potremmo misurare l'informazione che c'è in un libro? Ha a che vedere con il numero di pagine oppure sono importanti le parole? Un libro può avere più informazione di un altro? E se un libro fosse noioso e un altro particolarmente interessante? Un libro di 400 pagine che contenesse la frase "bla, bla, bla" avrebbe più o meno informazione di un elenco telefonico?
3. Basta spiegare che gli informatici misurano l'informazione considerando quanto è "sorprendente" un libro o un messaggio! Prendete come esempio qualcosa che già conoscete come il caso di un compagno che viene sempre a scuola a piedi. Se il compagno dice "lo sono venuto a scuola a piedi oggi" non dà nessuna informazione per il semplice fatto che non è sorprendente. Se il vostro compagno dicesse invece "oggi ho avuto un passaggio fino a scuola con un elicottero", quello sì che sarebbe sorprendente e fornirebbe un sacco di informazione.
4. Come può essere misurato il valore di sorpresa di un messaggio?
5. Un modo è verificare quanto sia difficile indovinare l'informazione. Se il compagno dicesse, "Indovina come sono venuto a scuola oggi" e fosse venuto a piedi, probabilmente potreste indovinarlo la prima volta. Ci potrebbero volere un po' più tentativi per arrivare all'elicottero e ancor di più se il compagno fosse venuto con una navicella spaziale.
6. La quantità di informazione che il messaggio contiene è misurata attraverso quanto sia facile o difficile indovinarla. Il gioco che segue ce ne dà un'idea.



"Grazie per il passaggio!"

Attività dell'indovina indovinello

Prima si sceglie un bambino. Gli altri bambini possono rivolgergli domande, ma questi potrà rispondere solo sì o no fino a che la risposta sia stata indovinata. Qualsiasi domanda è pertinente, purché la risposta sia strettamente 'sì' o 'no'.

Suggerimenti: State pensando a:

- ✓ un numero tra 1 e 100
- ✓ un numero tra 1 e 1000
- ✓ un numero tra 1 e 1,000,000.
- ✓ un qualsiasi numero intero
- ✓ una sequenza di 6 numeri con una logica, da indovinare in ordine dal primo all'ultimo (ad esempio 2, 4, 6, 8, 10)

Contate il numero di domande fatte. Questa è una misura del valore dell'*informazione*.

Discussione che segue

Quale strategia avete usato? Quale era la migliore?

Considerate che solo 7 domande possono essere poste per indovinare un numero tra 1 e 100 se dimezzate l'intervallo ogni volta. Per esempio:

è inferiore a 50? Sì.
è inferiore a 25? No.
è inferiore a 37? No.
è inferiore a 43? Sì.
è inferiore a 40? No.
è inferiore a 41? No.
Dev'essere 42! Sì!

Sorprendentemente, se l'intervallo numerico fosse incrementato fino a 1000, non ci sarebbe uno sforzo 10 volte maggiore ma sarebbero sufficienti solo 3 o 4 domande in più. Ogni volta che l'intervallo raddoppia si ha necessità di una domanda in più per trovare la risposta.

Un buon modo per approfondire l'argomento è quello di lasciare i bambini giocare a Mastermind.

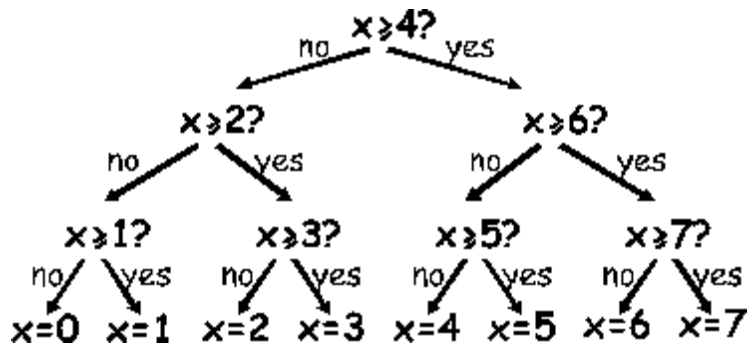
Estensione: Quanta informazione c'è in un messaggio? Gli informatici non hanno a che fare solo con i numeri ma anche con le lettere e possono quindi indovinare quale lettera è più probabile in una parola o in una frase.

Provate a giocare con un frase corta di 46 parole. Le lettere devono essere indovinate nell'ordine corretto, dalla prima all'ultima. Occorre qualcuno che scriva le lettere man mano che vengano indovinate e

tenga traccia di quanti tentativi occorrono per trovare ciascuna lettera. Qualsiasi domanda con sì/no può essere considerata. Un esempio potrebbe essere, "è una t?" "è una vocale?" "viene prima di m nell'alfabeto?" Lo spazio tra le lettere è esso stesso una lettera e può essere indovinata. Provate a verificare quali parti del messaggio sono più facili da indovinare.

Foglio di lavoro: alberi di decisione

Se già conoscete la strategia per fare domande, allora potete trasmettere un messaggio senza chiedere nulla. Ecco qui uno schema chiamato albero di decisione (decision tree) per indovinare i numeri tra 0 e 7:



Quante sono le decisioni yes/no (sì/no) per indovinare il numero 5?

Quante sono quelle necessarie per indovinare un numero qualsiasi?

Approfondite ora una caratteristica molto interessante.

Sotto i numeri 0, 1, 2, 3 nella riga finale dell'albero scrivete il numero in binario (Attività 1).

Guardate attentamente l'albero. Se no=0 e yes/sì=1 cosa ne ricavate?

Nel gioco del numero da indovinare cercate di scegliere domande tali che la sequenza di risposte che ne verrà rappresenti il numero esattamente in questo modo.

Scrivete il vostro albero di decisione per indovinare numeri tra 0 e 15.

Extra per gli esperti: Quale tipo di albero usereste per indovinare l'età di qualcuno?
E quale per indovinare la successiva lettera in una frase?

Cosa c'entra tutto questo?

Claude Shannon, che era un matematico americano molto famoso (oltre ad essere matematico era un grande saltimbanco e un ottimo mono-ciclista), fece molti esperimenti con questo gioco. Egli misurò la quantità di informazione in 'bits': ogni risposta sì/no è equivalente al bit 1/0. Così facendo scoprì che la quantità di "informazione" contenuta in un messaggio dipende da ciò che già si sa. Qualche volta possiamo fare una domanda che elimina la necessità di porne molte altre. In questo caso, il contenuto di informazione del messaggio è basso. Per esempio, se si lancia una volta una moneta l'informazione è normalmente di 1 bit: testa o croce. Ma se la moneta fosse truccata e testa uscisse 9 volte su dieci, allora l'informazione non sarebbe più 1 bit: ci si creda o no, sarebbe inferiore. Come si fa a sapere quale è il risultato del lancio della moneta con meno di una domanda sì/no? Semplice, basta usare domande come "i prossimi due lanci danno entrambi testa?" Per una sequenza di lanci con la moneta truccata, la risposta sarà "sì" circa l'80% delle volte. Sul 20% delle volte dove la risposta è "no," dovrete porvi altre due domande per trovare il caso esatto. Ma in media, avrete posto meno di una domanda per lancio di moneta!



Shannon chiamò con il termine "entropia" il contenuto informativo di un messaggio. L'entropia dipende non solo dal numero dei possibili esiti – nel caso di una moneta, due – ma anche dalla probabilità che ciò accada. Eventi improbabili o informazioni infrequenti hanno bisogno di molte più domande per far indovinare il messaggio perché questi portano molte informazioni che non si conoscevano – proprio come prendere un elicottero per andare a scuola.

L'entropia di un messaggio è molto importante per gli informatici. Non si può infatti comprimere un messaggio al di sotto della sua entropia in bit e il miglior sistema di compressione è equivalente al gioco delle domande. Poiché è un programma che pone le 'domande', la sequenza può essere riprodotta anche in un secondo momento e se si memorizzano le risposte (in bit) l'informazione può essere ricostruita! Il miglior sistema di compressione può ridurre file di testo a circa un quarto della dimensione originaria, un grosso risparmio di spazio!

Il metodo degli indovinelli può anche essere usato per costruire una interfaccia per computer che sia in grado di predire cosa l'utente digiterà! Questo è molto utile per persone con disabilità fisiche che abbiano difficoltà a digitare con la tastiera. Il computer suggerisce cosa probabilmente sarà digitato e l'utente sceglie la lettera corretta. Un buon sistema ha bisogno solo di due risposte sì/no per ciascun carattere che può essere di grande aiuto per coloro che hanno difficoltà nel compiere i movimenti fini che controllano il mouse o la tastiera. Un sistema analogo è usato ad esempio per digitare un testo SMS nei cellulari.

Soluzioni e suggerimenti

La risposta ad una domanda sì/no corrisponde esattamente ad un bit di informazione, sia con una domanda semplice "è più di 50?" o complessa "è tra 20 e 60?"

Nel gioco del numero da indovinare le domande sono scelte in un certo modo: la sequenza delle risposte è proprio la rappresentazione binaria del numero. Tre è 011 in binario ed è rappresentato dalle risposte "No, sì, sì" nell'albero di decisione e, come visto, possiamo scrivere no al posto di 0 e sì al posto di 1 e viceversa.

Un albero di decisione per l'età potrebbe avere a che vedere con numeri ancora inferiori di domande, poiché l'aspetto della persona è già un'informazione che abbiamo e che influenza il resto.

L'albero di decisione relativo alle lettere successive in una sequenza dipende sicuramente dalle lettere precedenti.

Parte II

**Far lavorare i computer — *gli*
*Algoritmi***

Far lavorare i computer

I computer agiscono seguendo una sequenza di istruzioni. Queste istruzioni permettono loro, ad esempio, di ordinare, trovare o trasferire informazioni. Per fare queste cose nel minor tempo possibile si ha la necessità di impiegare metodi efficienti sia per trovare quel che si cerca in grosse moli di dati sia per comunicare attraverso la rete.

Un algoritmo è un insieme di istruzioni preparato per completare un lavoro. L'idea di algoritmo è centrale per l'informatica. Gli algoritmi, infatti, sono il modo in cui noi diciamo ai computer di risolvere i problemi. Alcuni algoritmi sono più veloci di altri e la maggior parte degli algoritmi che sono stati scoperti hanno reso possibile risolvere problemi che prima richiedevano una incredibile quantità di tempo: per esempio, scoprire milioni di cifre del pi greco o tutte le pagine che contengono un dato nome nel World-Wide Web o trovare il miglior modo di stivare pacchi in un container, oppure capire se numeri molto grandi (100 cifre e più) sono primi.

Il termine "algoritmo" deriva dal nome di *Mohammed ibn Mūsā Al-Khowārizmī* che significa: *Mohammed*, figlio di *Moses*, da *Khwārezm*, che operò in un centro accademico noto come la Casa della Sapienza a Baghdad attorno all'800 dopo Cristo. I suoi lavori trasmisero l'arte Hindu del calcolo agli Arabi e in seguito all'Europa. Quando furono tradotti in Latino nel 1120 le prime parole del testo furono "Dixit Algorismi" così disse Algorismi".

Attività 6

Battaglia navale — *Algoritmi di ricerca*

Sommario

Ai computer si richiede spesso di cercare informazioni all'interno di grandi moli di dati. Per far ciò, essi necessitano di metodi veloci ed efficienti. L'attività presenta tre diversi tipi di ricerca: lineare, binaria e hash.

Abilità

- ✓ Ragionamento logico

Età

- ✓ A partire da 9 anni

Materiale

Ciascun bambino avrà bisogno di:

- ✓ una copia della battaglia navale
 - 1A, 1B per il gioco 1
 - 2A, 2B per il gioco 2
 - 3A, 3B per il gioco 3
- ✓ si potrebbe aver bisogno anche di qualche copia delle battaglie navali supplementari 1A', 1B', 2A', 2B', 3A', 3B'.

Battaglia navale

Attività introduttiva

1. Scegliete 15 bambini da allineare di fronte al resto della classe. Date a ciascun bambino una carta con un numero (in ordine casuale). Mantenete i numeri nascosti al resto della classe.
2. Date ad un altro bambino un contenitore con quattro o cinque caramelle. Il suo scopo è quello di indovinare un dato numero. Il bambino può “pagare” per sbirciare una carta particolare dei suoi compagni in fila. Se indovina il numero corretto prima di usare tutti i dolci che ha, allora può tenere quelli rimasti.
3. Ripetete i passi precedenti con altri bambini.
4. Adesso mescolate le carte e ridatele nuovamente ai bambini. Questa volta, fate in modo che i bambini si mettano in ordine ascendente. Ripetete la ricerca.

Se i numeri sono ordinati, una saggia strategia è quella di effettuare solo un “pagamento” per eliminare metà dei bambini poiché la carta di mezzo di fatto rivela anche i numeri delle carte di una metà della fila. Ripetendo questa strategia si dovrebbe trovare il numero cercato con sole 3 caramelle. L’incremento di efficienza sarà di conseguenza ovvio.

Attività

I bambini possono comprendere come un computer effettua una ricerca giocando a battaglia navale. Mentre giocano, fate in modo che pensino alle strategie che stanno usando per localizzare le navi.

Battaglia navale — *Un gioco a ricerca lineare*

Leggete le istruzioni che seguono ai bambini

1. Organizzatevi in coppie. Uno di voi ha il foglio 1A, l'altro l'1B. Non mostrate il vostro foglio al compagno!
2. Entrambi cerciate una delle 26 navi della parte alta del foglio e dite al compagno il numero. Quella sarà la vostra nave.
3. Adesso prendetevi tutto il tempo per indovinare dov'è la nave del compagno. (A turni uno dice la lettera della nave (da A a Z) e il compagno comunica il numero della nave che corrisponde a quella lettera nel suo schema. Se il numero corrisponde alla nave scelta dal compagno la nave è affondata se no è acqua). Usate lo schema nella parte bassa del foglio per barrare le varie navi ad ogni tentativo per evitare di fare richieste inutili (ad esempio chiedendo due volte la stessa lettera).
4. Di quanti colpi avete bisogno per localizzare la nave del compagno? Questo è il vostro punteggio di gioco: vince chi ha meno punti.

(I fogli 1A' e 1B' sono in più per quei bambini che vogliono giocare ancora o che "inavvertitamente" hanno visto i fogli dei compagni. I fogli 2A', 2B' e 3A', 3B' servono per giochi che saranno presentati tra poco).

Discussione che segue

1. Cosa rappresenta il punteggio?
2. Quale valore potrebbero avere il punteggio minimo e massimo? (Sono 1 e 26 rispettivamente, assumendo che i bambini non sparano alla stessa nave due volte. Questo metodo è chiamato "ricerca lineare" perché si ha la necessità di toccare tutte le posizioni una ad una, nel peggiore dei casi).

Battaglia navale — *Un gioco a ricerca binaria*

Istruzioni

Le istruzioni per questa versione del gioco sono le stesse di quella precedente tranne che i numeri delle navi sono in ordine ascendente nei fogli. Spiegate questo ai bambini prima dell'inizio.

1. Organizzatevi in coppie. Uno ha il foglio 2A, l'altro il 2B. Non mostrate il vostro foglio al compagno!
2. Entrambi cerciate una delle 26 navi della parte alta del foglio e dite al compagno il numero. Quella sarà la vostra nave.
3. Adesso prendetevi tutto il tempo per indovinare dov'è la nave del compagno. (A turni uno dice la lettera della nave (da A a Z) e il compagno comunica il numero della nave che corrisponde a quella lettera nel suo schema. Se il numero corrisponde alla nave scelta dal compagno la nave è affondata se no è acqua). Usate lo schema nella parte bassa del foglio per barrare le varie navi ad ogni tentativo per evitare di fare richieste inutili (ad esempio chiedendo due volte la stessa lettera).
4. Di quanti colpi avete bisogno per localizzare la nave del compagno? Questo è il vostro punteggio di gioco: vince chi ha meno punti.

Discussione che segue

1. Cosa rappresenta il punteggio?
2. Quale strategia hanno usato coloro che hanno fatto meno punti?
3. Quale nave dovrete scegliere per prima? (Quella di mezzo vi dice quale metà scartare e in quale metà è quella da cercare). Quale posizione si dovrebbe scegliere successivamente? (Di nuovo, la miglior strategia è quella di scegliere la nave nel mezzo della parte dove si sa che c'è quella da scovare).
4. Se si applica questa strategia, quanti colpi sono necessari per colpire la nave? (Cinque al massimo).

Questo metodo è chiamato "ricerca binaria" perché divide il problema sempre in due parti.

Battaglia navale — *Un gioco a ricerca*

Hash

Istruzioni

1. Si prende un foglio come nei giochi precedenti dicendo al compagno il numero della nave scelta.
2. In questo gioco troverete a quale colonna (da 0 a 9) la nave appartiene. Per far ciò occorre semplicemente sommare le cifre del numero della nave. L'ultima cifra della somma identifica la colonna. Per esempio, per localizzare la nave il cui numero è 2345 occorre sommare $2+3+4+5$, il cui risultato è 14. L'ultima cifra è 4, perciò la nave deve essere nella quarta colonna. Una volta identificata la colonna occorre trovare quale nave della colonna è quella giusta. Questa tecnica è chiamata "hash" (che in inglese suona come mischiare) perché le cifre sono schiacciate assieme.
3. Giocate usando questa nuova strategia. Si può sfruttare il medesimo foglio per molte volte, basta scegliere la nave da colonne diverse.

(Al contrario degli altri giochi, i fogli di riserva 3A' e 3B' devono essere usati in coppia, perché la distribuzione delle navi in colonna deve corrispondere).

Discussione che segue

1. Accumulate e discutete i punteggi come prima.
2. Quali navi sono molto semplici da trovare? (Quelle che sono da sole nella colonna). Quali navi è complesso trovare? (Quelle la cui colonna ne contiene molte)
3. Quale tra le strategie di ricerca viste è la più veloce? Perché?

Quali sono i vantaggi per ciascuna delle tre modalità di ricerca viste? (Il secondo modo è più veloce del primo, ma il primo non richiede che le navi siano in ordine. Il terzo modo è in genere più veloce degli altri due, ma qualche volta potrebbe essere veramente lento. Nel peggiore dei casi, se tutte le navi sono nella medesima colonna, il terzo metodo è pari al primo, in termini di velocità di ricerca).

Attività aggiuntive

1. I bambini possono disegnarsi da soli i fogli per la battaglia usando uno dei tre formati. Per il secondo gioco occorre che mettano le navi in ordine ascendente. Chiedete come possono far diventare il gioco Hash molto complesso. (Il caso più difficile si presenta quando tutte le navi sono nella medesima colonna). E come lo si può far diventare il più semplice possibile? (Occorre inserire lo stesso numero di navi in ciascuna colonna).
2. Cosa succede se la nave che si stava cercando non era dove ipotizzato? (Nel gioco lineare impiegherete 26 colpi per trovarla. In quello binario 5. quando si usa quello Hash il numero di colpi dipende da quante navi ci sono nella colonna).
3. Usando la strategia a ricerca binaria quanti colpi sono necessari per trovare una nave tra qualche centinaio (circa 6), qualche migliaio (circa nove), oppure un milione (circa diciannove)? (Attenzione che il numero di colpi aumenta molto lentamente al crescere del numero di navi. Solo un colpo in più è richiesto quando il numero delle navi raddoppia, perciò il numero di colpi è proporzionale al logaritmo del numero totale delle navi).

Le mie navi

Numero di colpi sparati:

9058	7169	3214	5891	4917	2767	4715	674	8088	1790	8949	13	3014
A	B	C	D	E	F	G	H	I	J	K	L	M
8311	7621	3542	9264	450	8562	4191	4932	9462	8423	5063	6221	2244
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Le tue navi

Numero di colpi sparati:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1A

Le mie navi

Numero di colpi sparati:

1630	9263	4127	405	4429	7113	3176	4015	7976	88	3465	1571	8625
A	B	C	D	E	F	G	H	I	J	K	L	M
2587	7187	5258	8020	1919	141	4414	3056	9118	717	7021	3076	3336
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Le tue navi

Numero di colpi sparati:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1B

Le mie navi

Numero di colpi sparati:

163	445	622	1410	1704	2169	2680	2713	2734	3972	4208	4871	5031
A	B	C	D	E	F	G	H	I	J	K	L	M
5283	5704	6025	6801	7440	7542	7956	8094	8672	9137	9224	9508	9663
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Le tue navi

Numero di colpi sparati:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

2A

Le mie navi

Numero di colpi sparati:

33	183	730	911	1927	1943	2200	2215	3451	3519	4055	5548	5655
A	B	C	D	E	F	G	H	I	J	K	L	M
5785	5897	5905	6118	6296	6625	6771	6831	7151	7806	8077	9024	9328
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Le tue navi

Numero di colpi sparati:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

2B

Le mie navi										Numero di colpi sparati:																																									
0		1		2		3		4		5		6		7		8		9																																	
A	9047	B	1829	C	3080	D	9994	E	5125	F	1480	G	8212	H	8051	I	1481	J	4712	K	6422	L	7116	M	8944	N	4128	O	6000	P	7432	Q	4110	R	9891	S	1989	T	2050	U	8199	V	4392	W	1062	X	2106	Y	5842	Z	7057

Le tue navi										Numero di colpi sparati:																																									
0		1		2		3		4		5		6		7		8		9																																	
A		B		C		D		E		F		G		H		I		J		K		L		M		N		O		P		Q		R		S		T		U		V		W		X		Y		Z	

3A

Le mie navi										Numero di colpi sparati:																					
0		1		2		3		4		5		6		7		8		9													
A	9308	E	6519	H	1524	K	4135	L	9050	O	4200	R	3121	V	2385	Y	1990	B	1478	I	8112	M	1265	S	9503	W	5832	Z	2502		
C	8417	F	2469	I	8112	J	2000	N	5711	P	7153	Q	6028	T	1114	U	7019	X	1917												
D	9434	G	5105	J	2000																										

Le tue navi										Numero di colpi sparati:																									
0		1		2		3		4		5		6		7		8		9																	
A		C				E		H		L		O		R		V		W		B				S		T		U		X		Y		Z	
B		D				F		I		M		P		S																					
						G		J		N		Q		U																					

3B

Le mie navi

Numero di colpi sparati:

6123	1519	9024	5164	2038	2142	7156	9974	9375	7104	1004	1023	5108
A	B	C	D	E	F	G	H	I	J	K	L	M
1884	3541	5251	4840	3289	3654	2480	5602	8965	4053	2405	2304	1959
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Le tue navi

Numero di colpi sparati:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1A'

Le mie navi

Numero di colpi sparati:

2387	9003	3951	5695	1284	4761	7118	1196	1741	3791	3405	3132	6682
A	B	C	D	E	F	G	H	I	J	K	L	M
9493	9864	7359	1250	7036	2916	7562	9299	8910	6713	5173	8617	4222
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Le tue navi

Numero di colpi sparati:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1B'

Le mie navi

Numero di colpi sparati:

28	▲	326	▲	943	▲	1321	▲	1896	▲	2346	▲	2430	▲	2929	▲	3106	▲	3417	▲	4128	▲	4717	▲	4915
A		B		C		D		E		F		G		H		I		J		K		L		M
5123	▲	5615	▲	6100	▲	7015	▲	7120	▲	7695	▲	7812	▲	8103	▲	8719	▲	9020	▲	9608	▲	9713	▲	9911
N		O		P		Q		R		S		T		U		V		W		X		Y		Z

Le tue navi

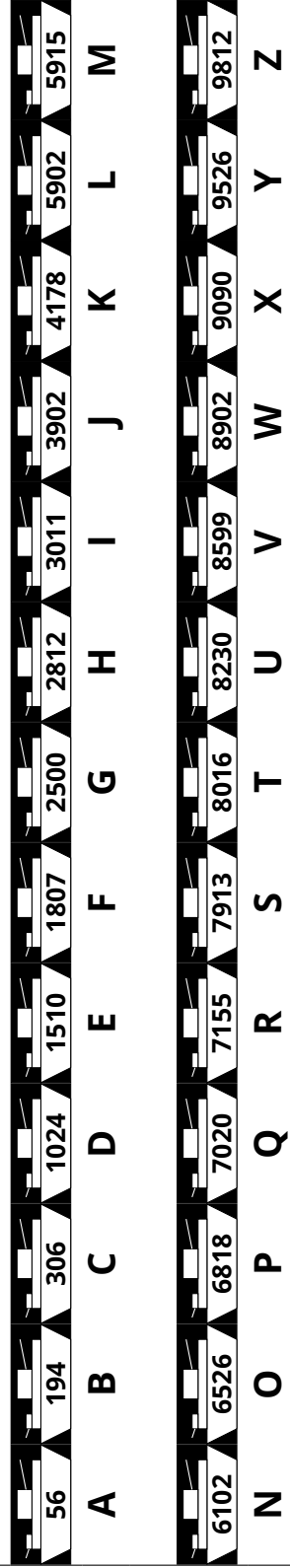
Numero di colpi sparati:

A		B		C		D		E		F		G		H		I		J		K		L		M
N		O		P		Q		R		S		T		U		V		W		X		Y		Z

2A'

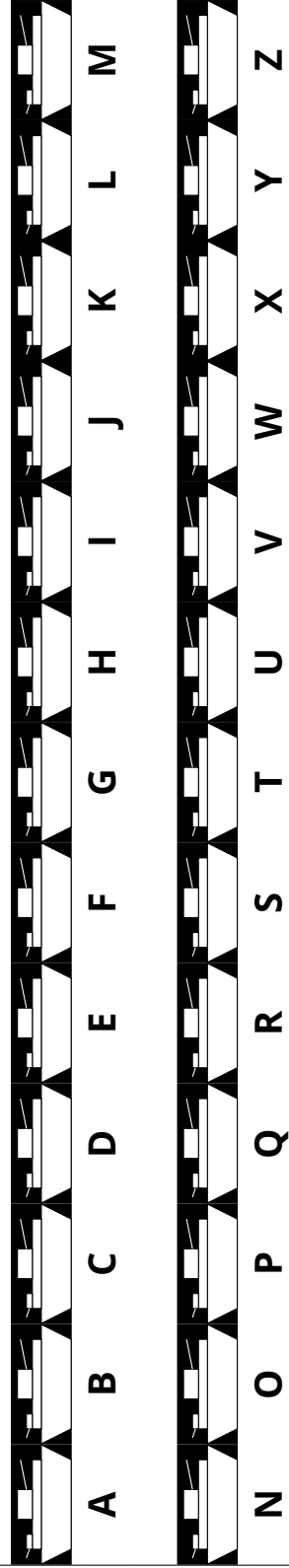
Le mie navi

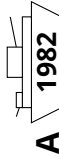
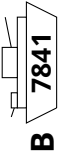
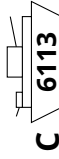
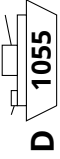

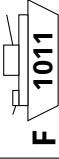
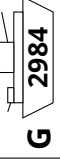



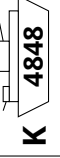


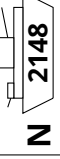

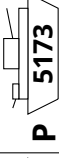
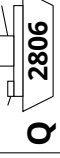
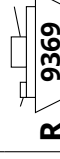
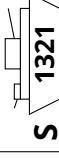
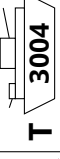
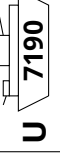


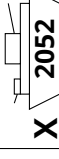
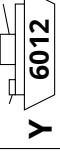
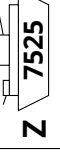
Numero di colpi sparati:

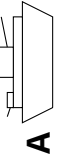



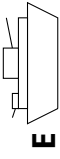


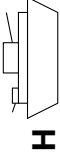


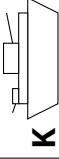
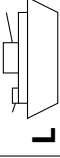


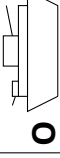


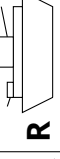


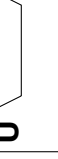
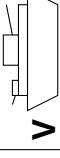


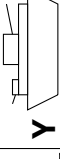
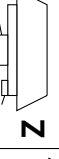


Le tue navi

Numero di colpi sparati:



Numero di colpi sparati:																												
Le mie navi																												
0	A 1982	B 7841	1	C 6113	D 1055	2	3	4	5	6	7	8	9															
																												

Numero di colpi sparati:																																			
Le tue navi																																			
0	A	B	C	D	1	E	F	G	2	H	I	J	3	K	4	L	M	N	5	6	O	P	Q	7	R	S	T	U	8	V	W	X	9	Y	Z
																																			

3A'

Le mie navi										Numero di colpi sparati:																									
0	8615	7003	1991	6211	1	1361	7644	5600	2	7726	9003	5557	3	3000	4	1814	2002	8844	5	6	9656	4002	1221	7	6993	3121	4300	1907	8	8208	9423	4176	9	2917	4122

Le tue navi										Numero di colpi sparati:																						
0			1			2	3	4					5				6				7					8		9				

3B'

Cosa c'entra tutto questo?

I computer memorizzano molte informazioni e hanno bisogno di rintracciarle spesso e velocemente. I motori di ricerca in Internet sono in grado di cercare informazioni all'interno di miliardi di pagine web in frazioni di secondo. Questa è sicuramente un'applicazione molto nota degli algoritmi di ricerca ma è al tempo stesso uno dei casi più complessi di uso, data la mole di informazioni da elaborare. I dati che un computer deve ricercare, come ad esempio una parola, un bar code o il nome di un autore, sono chiamati chiavi di ricerca (search key).

I computer riescono ad elaborare le informazioni molto velocemente. Si potrebbe pensare che scandiscano una dopo l'altra le informazioni dall'inizio della loro memoria continuando a cercare fintantoché l'informazione voluta non viene trovata. Questo, in effetti, è quello che fa il gioco a ricerca lineare (pag. 74), ma il metodo è molto lento anche per i computer. Si supponga per esempio che un supermarket abbia 10,000 prodotti differenti sugli scaffali. Quando un codice a barre è scannerizzato, il computer potrebbe dare un'occhiata a tutti quei 10,000 numeri per trovare il nome e il prezzo del prodotto. Anche se ciò comportasse solo un millesimo di secondo per ciascun prodotto, sarebbero necessari 10 secondi per passare l'intera lista. Immaginatevi quanto si impiegherebbe per passare alla cassa tutta la spesa della famiglia!

Una strategia migliore è la ricerca binaria (pag. 75). Con questo metodo i numeri sono messi in ordine. Il controllo di quello che sta in mezzo alla lista identificherà la metà nella quale la chiave di ricerca è presente. Il procedimento è ripetuto fintantoché l'articolo è trovato. Ritornando all'esempio del supermarket i 10,000 articoli possono essere ricercati solo con quattordici tentativi e con un tempo che si aggira su un centesimo di secondo, difficilmente percepibile.

Una terza strategia per cercare le informazioni è denominata hash (pag. 76). In questo caso la chiave di ricerca è manipolata per indicare esattamente dove si trova l'informazione cercata. Per esempio, se la chiave di ricerca è un numero telefonico, potreste sommare tutte le cifre del numero e calcolare il resto diviso 11. Ciò facendo si arriva grosso modo a quanto discusso nell'Attività 4: un piccolo frammento di dato il cui valore dipende da tutti gli altri. Di norma, il computer troverà velocemente quello che sta cercando senza alcun problema. Rimane comunque una piccola probabilità che molte chiavi di ricerca vadano a finire nella medesima locazione, nel qual caso il computer dovrà andare a fare una ricerca in quella lista finché non troverà l'informazione desiderata.

I programmatori di solito usano un qualche adattamento della strategia hash per la ricerca, a meno che non sia importante mantenere i dati in

ordine o a meno che una risposta lenta di tanto in tanto non sia accettabile.

Attività 7

Il piccolo e il grande — *Algoritmi di ordinamento*

Sommario

I computer sono spesso usati per ordinare un elenco di oggetti, come per esempio nomi in ordine alfabetico, appuntamenti o e-mail per data, numeri in ordine crescente o decrescente. Si è visto come l'ordinamento ci permetta di trovare facilmente l'oggetto che ricerchiamo facendo anche in modo di vedere a colpo d'occhio i valori estremi di qualsiasi lista. Per esempio, se si ordinano i risultati di un test allora il risultato più basso e quello migliore diventano evidenti.

Se si usa un metodo improprio, ci si può impiegare molto tempo per ordinare una lunga lista di oggetti, anche se si usa un computer veloce ed efficiente. Per fortuna, però, esistono molti metodi noti per effettuare tale operazione. In questa attività, i bambini scopriranno diversi modi per effettuare l'ordinamento e verificheranno come un buon metodo possa completare il compito molto più efficientemente e speditamente di un metodo semplice.

Abilità

- ✓ Usare bilance
- ✓ Ordinare
- ✓ Confrontare

Età

- ✓ A partire da 8 anni

Materiale

Ciascun gruppo di bambini avrà:

- ✓ insiemi di 8 contenitori della stessa dimensione ma con pesi diversi (ad esempio, cartoni del latte o scatole dei rotoli di pellicola o d'alluminio riempiti di sabbia)
- ✓ Bilance
- ✓ Foglio di lavoro: ordinare i pesi (pagina 96)
- ✓ Foglio di lavoro: divide et impera (pagina 97)

Il piccolo e il grande

Discussione

I computer hanno spesso la necessità di ordinare liste di oggetti. Allo stesso modo, per noi, pensare a tutti i luoghi ove riporre e ritrovare gli oggetti ordinati è altrettanto importante. Cosa accadrebbe se questi oggetti non fossero in ordine?

I computer solitamente confrontano solo due oggetti alla volta. L'attività alla pagina successiva usa questa restrizione per dare ai bambini l'idea di cosa significhi ordinare oggetti.

Attività

1. Dividete in gruppi di bambini.
2. Ciascun gruppo avrà bisogno di una copia della pagina dell'attività a pagina 96 oltre a pesi e bilance.
3. Lasciate che i bambini facciano l'attività, poi discutete assieme.

Attività del foglio di lavoro: ordinare pesi

Scopo: trovare il miglior metodo per ordinare un insieme di pesi.

Avrai bisogno di: sabbia o acqua, 8 contenitori identici, bilance.

Cosa fare:

1. Riempite ciascun contenitore con un diverso quantitativo di sabbia o acqua. Chiudete bene.
2. Mescolate i contenitori così che non si possa riconoscere il loro peso.
3. Trovate il contenitore più leggero.

Nota: Potete usare solo la bilancia per pesare ciascun contenitore. Solo due contenitori possono essere confrontati alla volta.

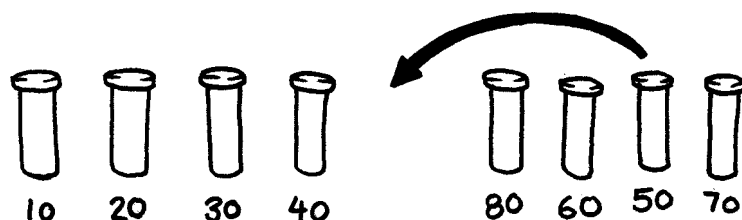
Qual è il sistema più facile di svolgere il compito?

4. Scegliete a caso 3 contenitori e metteteli in ordine dal più leggero al più pesante usando solamente la bilancia. Come potete fare? Qual è il minimo numero di confronti necessario? Perché?
5. Adesso mettetete tutti gli oggetti in ordine dal più leggero al più pesante.

Quando ritenete di aver concluso, controllate il risultato confrontando tra loro con la bilancia due oggetti vicini.

Selection Sort (ordinamento per selezione)

Un metodo che si può usare con il computer è chiamato selection sort. Funziona così: trovate per prima cosa l'oggetto più leggero e riponetelo da un lato. Successivamente, trovate il più leggero tra gli oggetti rimasti e mettetelo vicino a quello più leggero trovato precedentemente. Ripetete questa operazione fino a che non avete terminato gli oggetti da pesare.



Contate quanti confronti avete effettuato.

Extra per esperti: mostrate come potete calcolare matematicamente quanti passi sono necessari per ordinare 8 oggetti? E 9 oggetti? 20?

Attività del foglio di lavoro: divide et impera

Quicksort

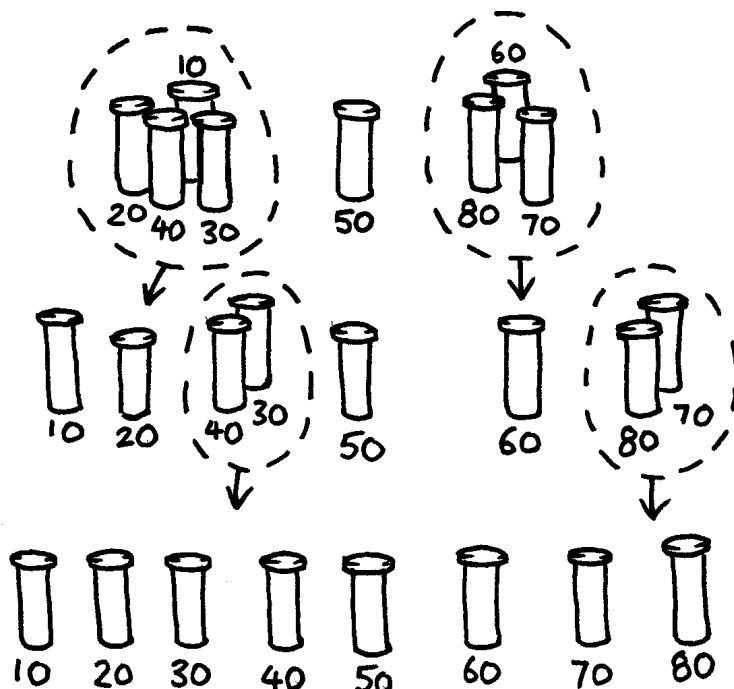
Quicksort è molto più veloce di selection sort, in particolare quando si hanno insiemi di oggetti molto grandi. Infatti, è uno dei migliori metodi di ordinamento conosciuti. Funziona così:

scegliete a caso un oggetto e posizionatelo su un piatto della bilancia.

Ora confrontate ciascun oggetto rimanente con quello precedentemente selezionato. Mettete quelli più leggeri a sinistra, l'oggetto selezionato per primo al centro e quelli più pesanti a destra. (Si potrebbero anche avere molti più oggetti da un lato rispetto all'altro).

Scegliete uno degli insiemi di oggetti, quello a destra o quello a sinistra e ripetete le precedenti operazioni. Fate lo stesso per l'altro insieme. Ricordatevi di mantenere al centro l'oggetto che selezionate per primo.

Continuate a ripetere questa procedura in tutti i gruppi fino a che ciascun gruppo conterrà un oggetto solo. A questo punto, gli oggetti saranno in ordine dal più leggero al più pesante.



Quanti confronti sono necessari in questo procedimento?

Quicksort è più efficiente di selection sort a meno che non accada di iniziare con l'oggetto più leggero o pesante in assoluto. Se siete così

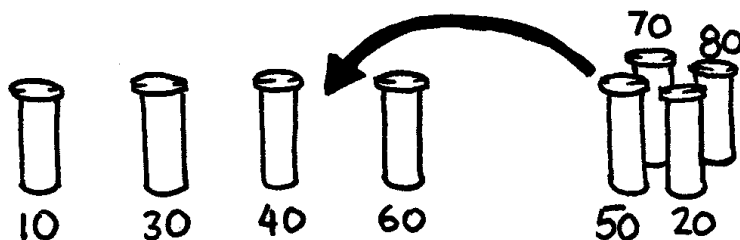
fortunati da aver scelto l'oggetto con il peso intermedio, dovrete aver effettuato solo 14 confronti, la metà di quelli effettuati con il selection sort. In ogni caso, quicksort non sarà peggiore di selection sort e, anzi, potrà essere molto molto più efficiente!

Extra per esperti: Se quicksort incidentalmente scegliesse sempre l'oggetto più leggero, quanti confronti effettuerebbe?

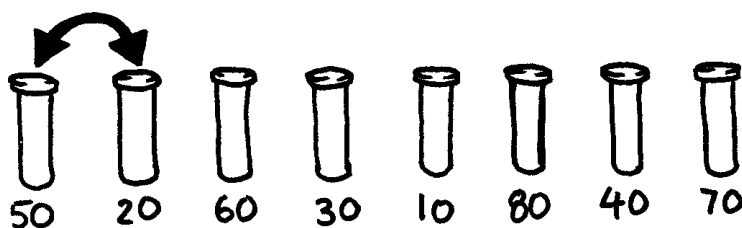
Variazioni ed estensioni

Sono stati inventati molti sistemi diversi per l'ordinamento.

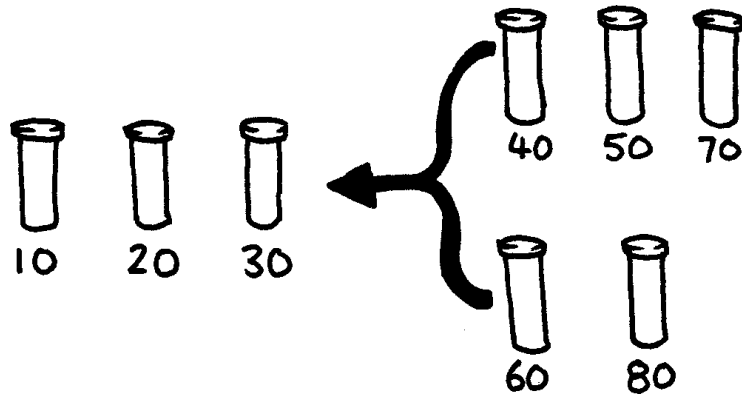
Insertion sort (ordinamento per inserimento) lavora rimuovendo ciascun oggetto da un insieme non ordinato e inserendolo nella posizione corretta in un altro insieme (come da figura). Ad ogni inserzione, l'insieme di oggetti non ordinati si riduce e quello ordinato cresce, fino a che l'intero insieme di oggetti originario è ordinato. I giocatori di carte usano molto spesso questo sistema per ordinare una *mano* di carte.



Bubble sort (ordinamento a bolle) opera nell'insieme di oggetti molte volte, scambiando ogni oggetto di posto, da destra a sinistra e viceversa quando incontra un oggetto a destra più leggero di quello di sinistra. L'insieme di oggetti è ordinato quando non sono più necessari scambi. Questo metodo non è molto efficiente, ma molti trovano che sia molto più facile da capire di altri.



Mergesort (ordinamento per fusione) è un altro metodo che usa 'divide et impera' per ordinare una lista di oggetti. Per prima cosa la lista è divisa casualmente in due liste con un numero di oggetti uguale (se gli oggetti sono in numero dispari, una delle due nuove liste conterrà un oggetto in più). Ciascuna delle due sotto-liste viene ordinata e alla fine entrambe vengono fuse assieme. Fondere le due sotto-liste è un procedimento semplice: basta rimuovere ripetutamente il più piccolo oggetto scelto tra i due all'inizio delle sotto-liste ordinate. Nella figura sottostante, gli oggetti con peso 40 e 60 grammi sono all'inizio delle due liste, così l'oggetto da prelevare e da fondere nella lista finale che si sta costruendo (a sinistra) è l'oggetto da 40 grammi. Come si ordinano le sotto-liste? Semplice, basta usare... Mergesort! Così facendo, alla fine tutte le sotto-liste saranno composte solo da un unico elemento, pertanto non ci si dovrà preoccupare di sapere quando finire...



Cosa c'entra tutto questo?

L'informazione è molto più facile da trovare in un insieme ordinato. Gli elenchi telefonici e gli indici dei libri usano l'ordine alfabetico. La vita sarebbe molto più complessa se non lo facessero...! Se una lista di numeri è ordinata, i casi agli estremi sono più facili da notare perché questi sono all'inizio o alla fine della lista. I casi duplicati sono anch'essi facili da notare, perché sono vicini tra loro.

I computer passano molto del loro tempo a mettere in ordine oggetti, per questo motivo gli informatici hanno cercato di trovare metodi veloci ed efficienti per farlo. Alcuni dei metodi più lenti, come insertion sort, selection sort o bubble sort, possono essere utili in situazioni speciali, ma sono generalmente usati quelli più veloci come quicksort perché sono molto veloci quando operano su enormi liste. Per esempio, per ordinare 100.000 elementi, il quicksort è normalmente 2 mila volte più veloce del selection sort. Se però vogliamo ordinare un milione di elementi diventa 20 mila volte più veloce. I computer spesso devono elaborare milioni di elementi (moltissimi siti web hanno milioni di clienti e ogni fotografia scattata con una fotocamera digitale ha milioni di pixel); la differenza fra i due algoritmi può corrispondere a un secondo di elaborazione o più di cinque ore per svolgere esattamente lo stesso compito. Non solo il ritardo nella risposta non sarebbe tollerabile, ma questo significherebbe anche che sarebbe stata usata 20 mila volte più elettricità (che non solo causa un maggiore impatto ambientale ma riduce anche la vita delle batterie nei dispositivi portatili. La scelta dell'algoritmo giusto o errato ha quindi serie conseguenze.

Quicksort sfrutta un concetto chiamato ricorsione. Questo significa che si continua a dividere una lista di oggetti in parti più piccole applicando lo stesso tipo di ordinamento alle nuove parti più piccole. Questo approccio in particolare è denominato divide et impera. La lista è ripetutamente suddivisa finché è sufficientemente piccola da conquistare (impera). Per quicksort, le liste sono suddivise finché non contengono un solo elemento. È banale ordinare un solo elemento! Sebbene sembri molto laborioso, questo metodo è incredibilmente più veloce di altri.

Soluzioni e suggerimenti

1. il modo migliore di trovare l'oggetto più leggero è quello di considerare tutti gli oggetti uno alla volta tenendo traccia del più leggero fino a quel momento. Quindi, confrontate due oggetti e tenete il più leggero, ripetendo il metodo fino a che tutti gli oggetti siano stati considerati.
2. Confrontate i pesi su una bilancia. Ciò può essere fatto facilmente con 3 pesate e qualche volta 2 se i bambini comprenderanno che il confronto è un operatore transitivo (cioè, se A è più leggero di B e B è più leggero di C, allora A dev'essere più leggero di C).

Esperti:

Ecco un suggerimento per calcolare il numero totale di passi di confronto che vengono effettuati da selection sort.

Per trovare il più piccolo di due oggetti effettuerete un solo confronto, per 3 oggetti ne effettuerete 2, 3 confronti per 4 oggetti e così via. Per ordinare 8 oggetti usando selection sort avrete bisogno di 7 confronti per trovare il primo oggetto (il più leggero), 6 per il secondo, 5 per il terzo e così via:

$$7 + 6 + 5 + 4 + 3 + 2 + 1 = 28 \text{ confronti}$$

n oggetti necessiteranno di $1 + 2 + 3 + 4 + \dots + (n - 1)$ confronti per essere ordinati.

Sommare questi numeri diviene più facile se si raggruppano diversamente.

Per esempio, per sommare i numeri $1 + 2 + 3 + \dots + 20$, raggruppateli così:

$$\begin{aligned} &(1 + 20) + (2 + 19) + (3 + 18) + (4 + 17) + (5 + 16) + \\ &(6 + 15) + (7 + 14) + (8 + 13) + (9 + 12) + (10 + 11) \\ &= 21 \times 10 = 210 \end{aligned}$$

In generale, sarà $1 + 2 + 3 + 4 + \dots + (n - 1) = \frac{n(n-1)}{2}$.

Attività 8

Batti il tempo — *Reti di ordinamento*

Sommario

Anche se i computer sono veloci, c'è un limite a quanto rapidamente riescono a risolvere i problemi. Un modo per accelerare ulteriormente il tempo di soluzione è quello di usare molti computer alla volta ciascuno dei quali risolve una parte del problema. In questa attività useremo le reti di ordinamento per capire come effettuare più di un confronto nello stesso istante.

Abilità

- ✓ Confrontare
- ✓ Ordinare
- ✓ Sviluppare algoritmi
- ✓ Risolvere problemi assieme

Età

- ✓ A partire da 7 anni

Materiale

Questa attività si può effettuare all'aperto.

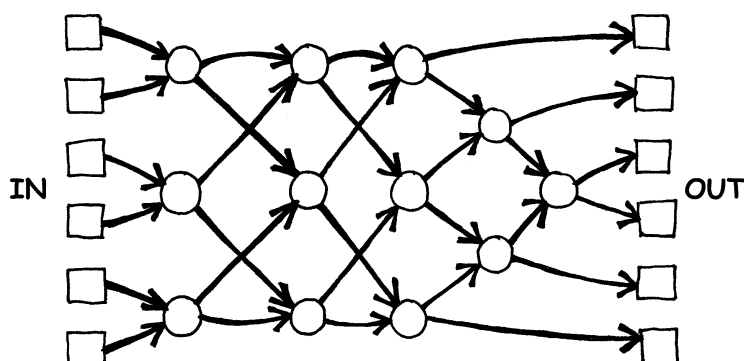
- ✓ Gesso
- ✓ Due insiemi di 6 carte.

Ricopiate il foglio principale: reti di ordinamento (pagina 107) in un foglio e ritagliatelo

- ✓ Un cronometro

Reti di ordinamento

Prima di iniziare l'attività usate il gesso per disegnare su una superficie esterna lo schema seguente:

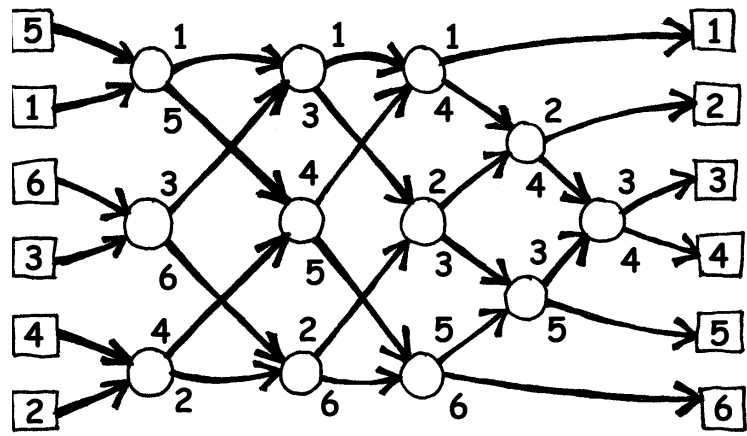


Istruzioni per i bambini

Questa attività vi mostrerà come i computer ordinano numeri casuali sfruttando la cosiddetta rete di ordinamento.

1. Organizzatevi in gruppi di 6. Solo un gruppo alla volta potrà giocare.
2. Ogni giocatore deve avere una carta numerata.
3. Ogni giocatore deve sostare nel quadrato a sinistra (IN) dello schema disegnato. I giocatori con i numeri devono essere mescolati.
4. Muovetevi lungo le linee. Quando raggiungete un cerchio **dovete attendere che arrivi un compagno.**
5. Quando il compagno arriva, confrontate le vostre carte. La persona con il numero più piccolo prosegue verso sinistra, l'altro verso destra.
6. Siete nell'ordine giusto quando arrivate all'altro capo dello schema?

Se una squadra compie un errore, i bambini devono re-iniziare il gioco. Controllate se avete ben compreso l'operazione da effettuare nei cerchi dello schema, dove il valore più piccolo prosegue a sinistra e il più grande a destra. Per esempio:



Foglio da fotocopiare: reti di ordinamento

1

2

3

4

5

6

156

221

289

314

422

499

Variazioni

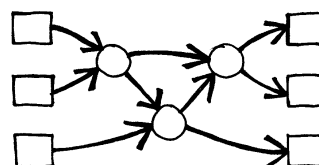
1. Quando i bambini hanno raggiunto familiarità con l'attività, usate il cronometro per misurare quanto tempo ciascuna squadra impiega per concludere il gioco.
2. Usate carte con numeri più grandi (ad esempio, quelli presenti nella fotocopia principale).
3. Ideate carte con numeri ancora più grandi che siano complesse da confrontare, oppure usate parole e confrontatele alfabeticamente.
4. Non è necessario confrontare numeri. Nell'ambito di un esercizio multidisciplinare, per esempio con il corso di musica, possono essere confrontate note su un pentagramma che possono essere ordinate a seconda del loro tono, dalla più grave alla più acuta o in ordine di durata.

Attività aggiuntive

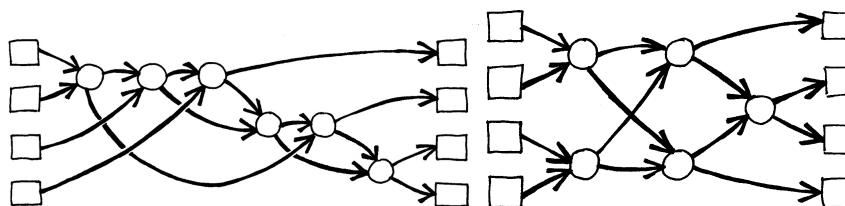
1. Cosa succede se il numero più piccolo va a destra invece che a sinistra e viceversa? (I numeri saranno ordinati in ordinamento inverso)

Lo schema funziona lo stesso se si va al contrario? (Non funziona e i bambini dovrebbero essere in grado di trovare un esempio di input che non viene ordinato correttamente)

2. Cercate di progettare una rete di ordinamento più piccola o più grande. Per esempio, ecco una rete che ordina solo 3 numeri. I bambini dovrebbero riuscire ad idearla da soli.

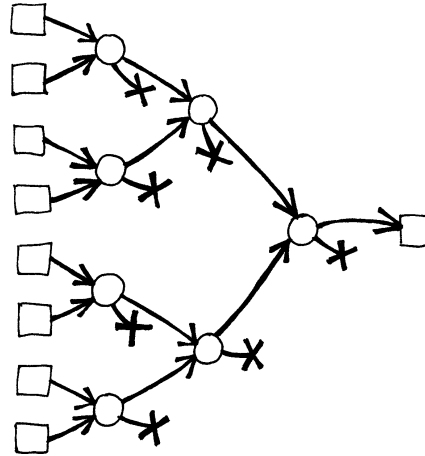


3. Seguono due diverse reti di ordinamento che effettuano l'operazione su 4 numeri. Quale delle due è più veloce? (La seconda. La prima richiede che tutti i confronti siano effettuati in sequenza, uno dopo l'altro, mentre la seconda prevede che qualche operazione sia effettuata contemporaneamente. La prima rete è un esempio di calcolo seriale, la seconda sfrutta il calcolo parallelo per andare più veloce).



4. Cercate di progettare una rete ancora più grande..

5. Le reti possono anche essere usate per trovare il valore minimo dato in input. Per esempio, a destra è disegnata una rete per 8 valori: l'unico valore di uscita sarà quello minimo (gli altri valori rimarranno intrappolati nella rete).



6. Quali processi di tutti i giorni potrebbero o non potrebbero essere accelerati usando il parallelismo? Per esempio, cucinare un pasto sarebbe più lento se usassimo solo un fornello perché tutti i piatti dovrebbero essere cotti uno dopo l'altro. Quali lavori potrebbero essere completati più velocemente impiegando più persone? E quali no?

Cosa c'entra tutto questo?

Via via che si usano sempre più i computer si vorrebbe che riuscissero ad elaborare sempre più velocemente le informazioni.

Un modo per incrementare la velocità di elaborazione è quello di scrivere programmi che usano meno passi per risolvere il problema (come mostrato nelle attività 6 e 7).

Un altro modo per risolvere i problemi più velocemente è quello di usare più computer nello stesso tempo, operanti su diverse parti dello stesso problema. Per esempio, nella rete di ordinamento con 6 numeri, sebbene siano necessari 12 confronti per ordinare i numeri, si potranno effettuare fino a 3 comparazioni simultaneamente. Ciò significa che il tempo richiesto è solo di 5 passi di elaborazione. Questa rete parallela ordina la lista di numeri ad una velocità più che doppia rispetto ad un sistema che effettua un confronto per volta.

Non tutti i problemi possono essere risolti più velocemente usando il calcolo parallelo. Per analogia, immaginate una persona che faccia un fossato lungo 10 metri. Se 10 persone facessero un metro di quel fossato, l'intero lavoro sarebbe terminato più velocemente. Ma lo stesso non accadrebbe con uno scavo di 10 metri di profondità - il secondo metro non sarebbe accessibile finché non fosse terminato il primo. Gli informatici stanno ancora studiando qual è il miglior modo di suddividere i problemi così che possano essere elaborati da computer in parallelo.

Attività 9

La città fangosa — *Minimal Spanning Tree*

Sommario

La nostra società ha molti collegamenti in rete: la rete telefonica, la rete energetica, la rete stradale. Per una rete in particolare, ci sono solitamente più scelte su come posizionare gli elementi (le strade, i cavi o i collegamenti radio). Dobbiamo quindi trovare i modi più efficienti per collegare gli oggetti tra loro.

Abilità

- ✓ Risolvere problemi

Età

- ✓ A partire da 9 anni

Materiale

Ogni bambino avrà bisogno di:

- ✓ Fotocopia del Foglio di lavoro: il problema della città fangosa (pagina 113)
- ✓ Tavole o quadrati di cartone (circa 40 per bambino)

La città fangosa

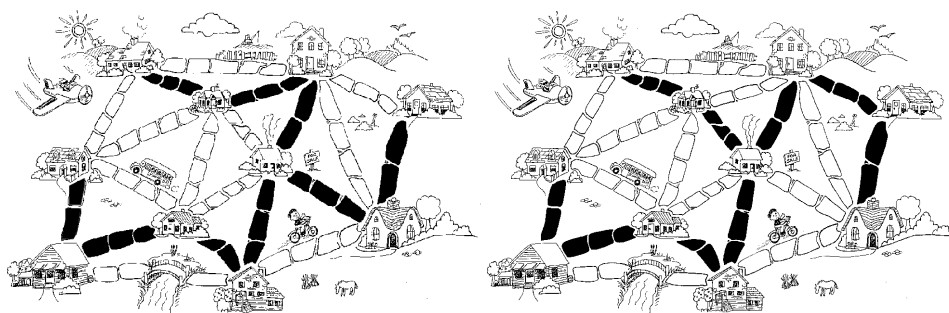
Introduzione

Questa attività vi mostrerà come sono usati i computer per trovare la miglior soluzione in alcuni problemi quotidiani, come ad esempio quello di collegare linee elettriche tra case. I bambini devono usare il foglio di lavoro di pagina 93 che spiega il problema della 'Città Fangosa'.

Discussione che segue

Condividete la soluzione trovata dai bambini. Quale strategia hanno usato?

Una buona strategia di partenza è quella di partire con una mappa vuota e gradualmente posizionare le tavole finché tutte le case sono collegate tra loro, a partire dai collegamenti di lunghezza inferiore, facendo attenzione di non collegare case già tra loro collegate. Si possono avere diverse soluzioni se si scelgono diversi cammini di collegamento della stessa lunghezza. Due soluzioni alternative sono illustrate di seguito.



Un'altra strategia è quella di cominciare con tutti i cammini già selezionati rimuovendoli man mano, se non sono necessari. In ogni caso, ciò richiede maggior sforzo della strategia precedente.

Dove potete trovare reti di questo tipo nella quotidianità?

Gli informatici chiamano la rappresentazione di queste reti con il nome di "grafi". Le reti reali possono essere rappresentate da grafi per arrivare a risolvere problemi come ad esempio progettare il posizionamento migliore per una rete di strade tra città, oppure trovare il miglior modo di connettere delle città con voli aerei.

Ci sono anche molti algoritmi che possono essere applicati ai grafi, come ad esempio quello di trovare la distanza inferiore tra due punti o il cammino più corto che tocca tutti i punti del grafo

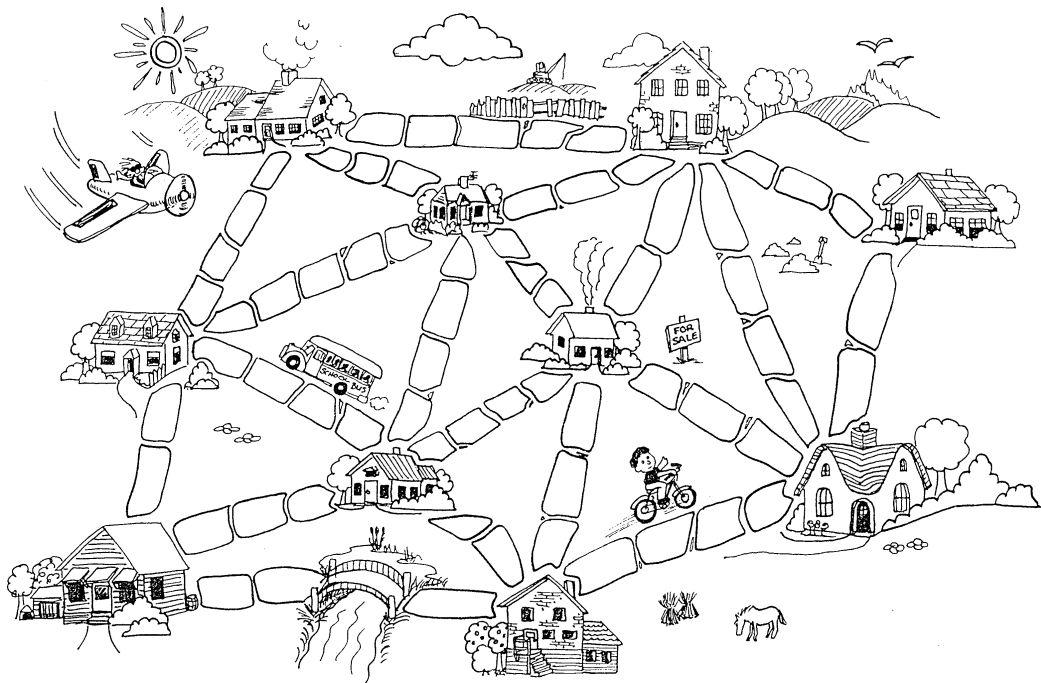
Foglio di lavoro: Il Problema della Città Fangosa

Tanto tempo fa c'era una città che non aveva strade. Gironzolare nella città era particolarmente difficile dopo un temporale perché tutt'intorno c'era solo fango. Le macchine si bloccavano nel fango e le persone si sporcavano sempre gli stivali. Il sindaco della città decise così che qualche strada doveva essere pavimentata ma non voleva spendere più soldi del necessario perché voleva costruire anche una piscina. Il sindaco specificò allora due condizioni

1. devono essere pavimentate abbastanza strade da rendere possibile per ognuno andare dalla propria casa ad un'altra qualsiasi casa,
2. la pavimentazione deve costare il meno possibile.

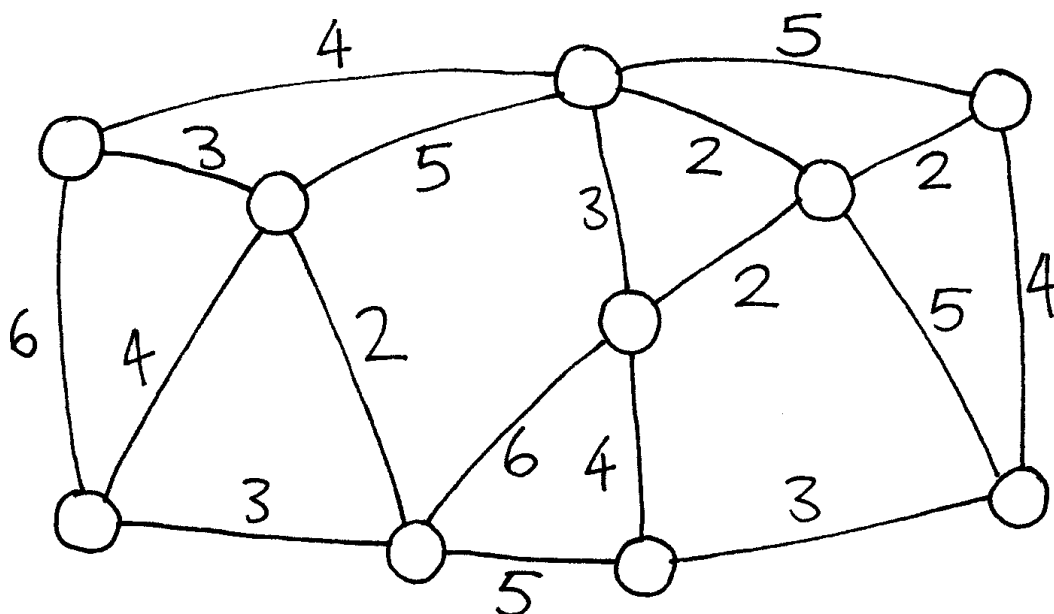
Quella che segue è la mappa della città. Il numero di tavole (pietre) tra le case rappresenta il costo per pavimentare quel tratto. Trovate il miglior percorso per connettere tutte le case, ma usate il minor numero di tavole (pietre) possibile (il ponte non conta, non ha necessità di essere pavimentato).

Quale strategia usereste per risolvere il problema?



Variazioni ed estensioni.

Quello che segue è un modo alternativo di rappresentare città e strade:



Le case sono rappresentate da cerchi, le strade fangose da linee e la lunghezza delle strade da un numero vicino alla linea. .

Gli informatici e i matematici usano spesso questo diagramma per rappresentare problemi. Lo chiamano *grafo*. Questo potrebbe causare un po' di confusione poiché la parola assomiglia a quella usata in ambito statistico per rappresentare dati numerici cioè il "*grafico*", ma sono due cose diverse, anche se i nomi si assomigliano. In un grafo, le lunghezze delle linee possono non essere correlate al numero che hanno vicino.

Costruite un vostro esempio di città fangosa e risolvetele con i vostri compagni.

Potete trovare una regola che descriva quante strade o connessioni sono necessarie per la miglior soluzione? Dipende da quante case ci sono nella città?

Cosa c'entra tutto questo?

Supponete di dover progettare come portare ad una nuova comunità l'energia oppure il gas o l'acqua. Una rete di fili o di tubi deve connettere tutte le case all'azienda elettrica o di fornitura dell'acqua. Ciascuna casa deve essere collegata alla rete in un qualche punto e il percorso di collegamento tra la casa e l'azienda non è importante più di tanto, ciò che è importante è che la casa sia collegata.

Il problema di progettare una rete con un percorso totale minimo è chiamato *minimal spanning tree*.

I *minimal spanning tree* non sono utili solo per il gas o l'elettricità; ci aiutano anche a risolvere problemi nelle reti di computer, nelle reti telefoniche, nelle condotte petrolifere, nelle rotte aeree. Ad esempio, quando si decide quale rotta di viaggio è la migliore per una persona, occorre considerare non solo quanto costerà al viaggiatore, ma anche quanto costerà in generale. Nessuno vorrebbe passare ore in aereo prendendo la rotta più lunga perché è più economica. Il problema della città fangosa potrebbe non essere di molto aiuto per queste reti, perché calcola il percorso minimo totale delle strade o dei voli aerei.

I *Minimal spanning tree* sono utili anche come primo passo per risolvere altri problemi come, ad esempio il "problema del commesso viaggiatore", che cerca di trovare il percorso più corto per visitare ogni punto di una rete.

Ci sono algoritmi efficienti per risolvere i problemi *minimal spanning tree*. Un metodo semplice che conduce ad una soluzione ottima è quello di partire senza alcun collegamento, aggiungendoli a partire dai più corti e connettendo solo parti della rete che non sono ancora state connesse. Questo è chiamato algoritmo di Kruskal (J.B. Kruskal lo pubblicò nel 1956).

Per molti problemi sui grafi, incluso il "problema del commesso viaggiatore", gli informatici devono ancora trovare metodi sufficientemente veloci per la miglior soluzione possibile.

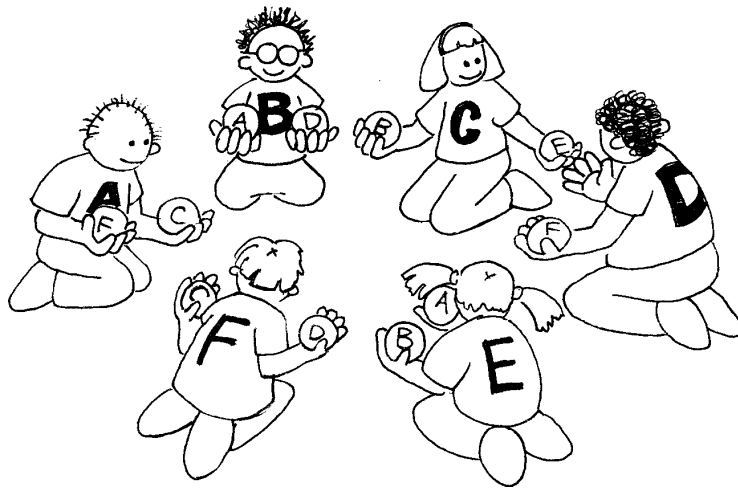
Soluzioni e suggerimenti

Variazioni ed estensioni (pagina 114)

Quante strade o collegamenti sono necessari se ci sono n case in città?
Ne deriva che una soluzione ottima avrà esattamente sempre $n - 1$ collegamenti, perché ciò è sufficiente per collegare n case e un collegamento in più creerebbe rotte alternative tra di esse.

Attività 10

Il gioco dell'arancia — *Instradamento e deadlock nelle reti.*



Sommario

Quando ci sono molte persone che devono usare insieme una risorsa (per esempio le automobili che usano una strada o i messaggi in Internet) c'è la possibilità che le persone si blocchino a vicenda. Si dice in questo caso che il sistema è in *deadlock* (traducibile in italiano come "stallo"). Occorre che le persone (o i computer) collaborino attivamente perché questo non succeda.

Competenze richieste

- ✓ Soluzione cooperativa di problemi
- ✓ Capacità di ragionamento logico

Età

- ✓ A partire da 9 anni

Materiale

Ogni studente deve avere:

- ✓ Due arance o palle da tennis contrassegnate con la stessa lettera o due frutti di tipo diverso per ogni studente.
- ✓ Etichette o targhette col nome o capelli colorati, per creare una corrispondenza fra ogni studente e una lettera o un tipo di frutta.

Il gioco dell'arancia

Introduzione

Questo è un esempio di soluzione cooperativa di un problema. Si lavora a gruppi, ogni studente sarà contrassegnato da una lettera. Lo scopo è che ogni studente alla fine abbia le arance che recano l'etichetta con la propria lettera.

1. Dividere gli studenti in gruppi di cinque o più.
2. A ogni studente deve essere attribuita una lettera (unica all'interno del gruppo) che lo studente scriverà su una etichetta da tenere ben visibile (per esempio attaccandola alla maglietta). Se ci sono si possono usare i porta cartellini di riconoscimento. Preparate per ogni studente due arance con l'indicazione della lettera relativa. Uno studente solo per gruppo deve avere un'arancia con la propria lettera invece che due (le arance verranno distribuite agli studenti una per mano, nel gruppo dovrà rimanere esattamente una mano libera).
3. Distribuite ora casualmente le arance agli studenti del gruppo. Ogni studente avrà quindi due arance tranne uno che avrà una mano libera. Fate in modo però che nessuno studente abbia arance con la sua lettera.
4. Gli studenti devono passarsi le arance fino a quando non si giunga alla soluzione, cioè che ogni studente abbia le arance corrispondenti alla propria lettera. Occorre seguire queste regole.
 - a) Non si può tenere più di un'arancia in ogni mano.
 - b) Un'arancia può essere passata nella mano libera dello studente immediatamente alla propria sinistra o alla propria destra nel cerchio. (Lo studente può passare una qualsiasi delle proprie arance al vicino).

Gli studenti impareranno presto che se tenteranno di usare politiche *greedy* (tenendo le arance non appena avranno quella corrispondente alla propria lettera) allora il gruppo potrà non riuscire ad arrivare alla soluzione del problema. Greedy letteralmente significa "avido, ingordo". Nell'inglese informatico si chiamano metodi di soluzione greedy quelli che tentano di trovare la soluzione in modo incrementale passando per soluzioni parziali che, una volta trovate, non vengono più messe in discussione. Non sempre i metodi greedy raggiungono la soluzione o non trovano la soluzione migliore.

È necessario forse sottolineare che non vincono il gioco i singoli studenti quando hanno le arance con la propria lettera ma vince il gioco (e risolve il problema) il gruppo quando tutti hanno le arance corrette.

Discussione sul problema

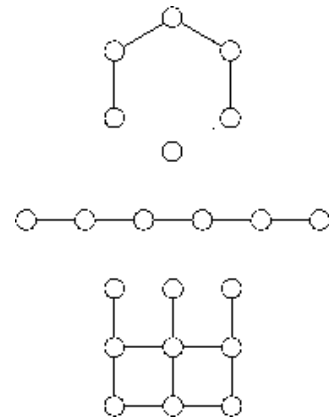
Quali strategie sono state usate dagli studenti per risolvere il problema?

Quando nella vita reale siete rimasti vittime di un deadlock? (Alcuni esempi possono essere ingorghi stradali, uso delle porte girevoli con molte persone che vogliono entrare e molte che vogliono uscire, ...)

Possibili estensioni

Provate l'attività con gruppi più o meno ampi.

- Chiedete agli studenti di inventare e provare regole diverse.
- Fate in modo che gli studenti tentino di risolvere il problema senza comunicare, magari seguendo una regola che hanno deciso a priori.
- Provate diverse configurazioni, come essere disposti in fila o in altre disposizioni dove alcuni studenti abbiano più di due vicini (come si può vedere nella figura qui a lato).



Cosa c'entra tutto questo?

problemi come l'instradamento e il deadlock sono presenti in molte reti, come le reti viarie, telefoniche o le reti informatiche come Internet. I tecnici spendono molto tempo per capire come risolvere questi problemi e come progettare le reti in modo che questi problemi siano facilmente risolvibili.

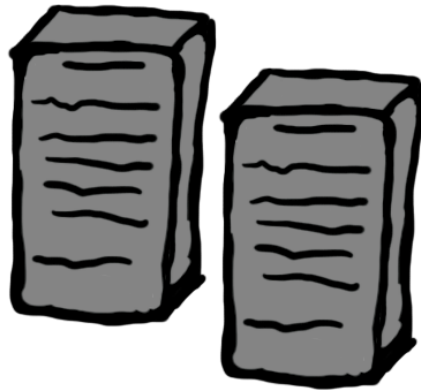
L'instradamento errato, le congestioni e il deadlock possono causare problemi frustranti in molte reti. Pensate solamente al traffico nelle ore di punta. A New York è successo molte volte che il traffico fosse così congestionato da finire in un vero deadlock: nessuno era più in grado di muovere la propria auto! Talvolta i computer sono "giù" nelle aziende, nelle banche o negli uffici pubblici a causa di un deadlock nelle comunicazioni. Progettare le reti in modo che l'instradamento delle comunicazioni sia efficiente e veloce e in modo da minimizzare le congestioni è un problema difficile per ingegneri di vari settori.

Talvolta più di una persona vuole lo stesso dato allo stesso tempo. Se uno specifico dato (come il saldo di un conto bancario) deve essere aggiornato, è importante bloccarlo (operazione di lock) durante l'aggiornamento. Se non viene bloccato, un altro utente potrebbe tentare di aggiornarlo nello stesso tempo e il risultato finale potrebbe essere scorretto. Se invece il blocco richiesto da un programma interferisce con il blocco richiesto da un altro programma ci può essere deadlock. Nel caso del conto bancario pensiamo a due operazioni contemporanee, la prima trasferisce il denaro dal conto A al conto B e la seconda dal conto B al conto A, quindi entrambe devono bloccare sia il conto A sia il conto B. Se il primo programma blocca il conto A mentre il secondo blocca il conto B il sistema è in deadlock perché il primo programma aspetterà che il secondo rilasci il blocco sul conto B mentre il secondo per farlo sta aspettando che il primo lasci libero il conto A.

Uno degli sviluppi più interessanti nella evoluzione della struttura dei computer è l'avvento dei computer paralleli, dove centinaia o migliaia di processori lavorano insieme collegati da una rete per formare un solo potentissimo computer. Molti problemi come quello del gioco dell'arancia devono essere risolti continuamente su queste reti (ma molto più velocemente!) per poter far funzionare questi computer.

Attività 11

Le Tavolette di Pietra — *Protocolli di comunicazione*



Sommario

Tutti i computer dialogano tra loro in Internet scambiandosi messaggi. Ma Internet non è affidabile e, di tanto in tanto, parte di questi messaggi viene perduta. Per ovviare a questo inconveniente e assicurarci che tutto vada a buon fine, possiamo aggiungere alcuni bit di informazione al messaggio originale. Questa informazione crea un "protocollo".

Abilità

- ✓ Problem solving cooperativo
- ✓ Ragionamento logico

Età

- ✓ Dai 9 anni in su

Materiali

Ogni studente avrà bisogno di:

- ✓ Diverse "Tavolette" su cui scrivere (v. pag. 125)

Ogni messaggero avrà bisogno di:

- ✓ Un po' di "carte delle operazioni" da svolgere con le Tavolette (v. pag. 124)

L'insegnante avrà bisogno di:

- ✓ Un timer

Le Tavole di Pietra

Introduzione

In questa attività gli studenti osservano come diversi metodi di comunicazione agiscano per arrivare a concludere positivamente le operazioni. Attraverso un confronto diretto sulle regole e procedure, gli studenti sono via via introdotti ai protocolli di comunicazione. Attraverso il gioco di ruolo, gli alunni verificano i propri protocolli in uno scenario non affidabile simile a quello che avviene in internet tramite lo scambio di pacchetti, specificatamente il protocollo TCP/IP.

Preparazione (30 minuti)

1. Per prima cosa si preparano le carte. Ci sarà bisogno di stampare le carte delle operazioni (sotto) e poi di ritagliarle. Queste sono alla base del gioco.
2. Dopo, si stabilisce quali messaggi far spedire dagli studenti. È importante che non siano frasi in italiano o parole da cui si possa risalire alla loro struttura originaria. Dev'essere qualcosa del tipo "1LHC255HD(RLLS" oppure un numero telefonico.
3. Occorre stampare anche qualche copia delle "Tavole". Ciascuna Tavola ha spazio per soli 6 caratteri o numeri, così che non si possa scrivere l'intero messaggio in una sola di esse. Ci sarà bisogno di circa 30 Tavole per ciascuno studente, a seconda di quanto si vuole far durare il gioco.

Nota: Le carte delle operazioni sono di tre tipi: ritarda, non spedire, spedisce. La qualità dei messaggi è determinata dal rapporto numerico tra queste carte. Più carte "spedisce" ci sono, più è affidabile il messaggio. Se, al contrario, ci sono più carte "ritarda" o "non spedire" allora significa che la rete è meno affidabile. Le carte agiscono in analogia a quanto accade in un canale di comunicazione o in una rete di computer.

Il gioco

1. Dividete la classe in coppie di alunni. È essenziale che ciascun membro della coppia sia distanziato dall'altro in modo che non possano né vedersi né comunicare. Due stanze sarebbero l'ideale, ma far sedere gli alunni ai lati opposti di una stessa classe è sufficiente.
2. Date ad un membro della coppia un messaggio da spedire al partner.
3. Mescolate le carte delle operazioni e scegliete un messaggero. L'insegnante può agire da messaggero, oppure il messaggero può essere uno studente nel caso di numero dispari di alunni. Si può prevedere più d'un messaggero nel caso di classi numerose.

4. Uno studente deve scrivere sulla Tavoletta e consegnarla al messaggero. La Tavoletta deve riportare almeno il nome dell'altra persona.
5. Il messaggero prende una carta dal mazzo ed effettua l'azione che vi è riportata.
6. Si ripetono i passi 4 e 5 per ciascuna Tavoletta

Dopo cinque minuti circa di caos e frustrazione, gli studenti dovrebbero capire che il solo nome non è sufficiente per un protocollo di comunicazione. A quel punto, si ferma la classe e si inizia una discussione su questo aspetto... Qual è la prima questione che sorge? È l'ordine delle Tavolette? Forse è il caso di utilizzare uno di quei 6 spazi su cui scrivere per indicare il numero della Tavoletta? Questo significa che ci sarà meno spazio per i veri e propri dati da comunicare e quindi cosa significa in termini di numero di Tavolette che devono essere utilizzate?

Dopo un altro po' di tempo sorgeranno altre questioni di cui discutere. I possibili problemi potrebbero essere Tavolette perdute o mancanti, oppure la non conoscenza se le Tavolette sono state effettivamente consegnate o meno e se devono essere rispedite.

Le soluzioni da suggerire potrebbero essere quelle di spedire al mittente informazioni di recapito avvenuto, attendendo che queste effettivamente arrivino prima di ripeditare ancora le precedenti Tavolette.

Ciò significa che anche gli studenti destinatari dei messaggi devono avere a disposizione Tavolette su cui scrivere per spedire messaggi a loro volta, accordandosi sui caratteri da inserire in quei 6 spazi prima di ricominciare il gioco.

Saranno necessari almeno due studenti per il gioco, ma è raccomandabile che siano il maggior numero possibile. Se la classe è particolarmente numerosa, è meglio prevedere più di un messaggero. Ancora una volta, si introduca questa discussione... Cosa succede se sono presenti molti messaggeri? E se ce ne fosse solo uno?

Consegna questa tavoletta ora	Consegna questo messaggio dopo il prossimo
Consegna questa tavoletta ora	Consegna questo messaggio dopo il prossimo
Consegna questa tavoletta ora	Consegna questo messaggio dopo il prossimo
Consegna questa tavoletta ora	Non consegnare questo messaggio
Consegna questa tavoletta ora	Non consegnare questo messaggio

<p>Destinatario:</p> <table border="1" data-bbox="300 338 756 409"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>Mittente:</p>							<p>Destinatario:</p> <table border="1" data-bbox="815 338 1272 409"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>Mittente:</p>						
<p>Destinatario:</p> <table border="1" data-bbox="300 660 756 732"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>Mittente:</p>							<p>Destinatario:</p> <table border="1" data-bbox="815 660 1272 732"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>Mittente:</p>						
<p>Destinatario:</p> <table border="1" data-bbox="300 983 756 1055"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>Mittente:</p>							<p>Destinatario:</p> <table border="1" data-bbox="815 983 1272 1055"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>Mittente:</p>						
<p>Destinatario:</p> <table border="1" data-bbox="300 1305 756 1377"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>Mittente:</p>							<p>Destinatario:</p> <table border="1" data-bbox="815 1305 1272 1377"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>Mittente:</p>						
<p>Destinatario:</p> <table border="1" data-bbox="300 1628 756 1700"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>Mittente:</p>							<p>Destinatario:</p> <table border="1" data-bbox="815 1628 1272 1700"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> <p>Mittente:</p>						

Le Tavolette di Pietra

In un'antica città ci sono Governanti molto potenti. Questi Governanti decidono sulla vita della città e prendono decisioni molto importanti. Ciascuno di essi abita in una casa diversa della città.

I Governanti devono comunicare molto spesso e hanno la necessità di spedire e ricevere messaggi da un capo all'altro della città. I Governanti si identificano tra loro in base al numero della loro casa e hanno la possibilità di utilizzare un gruppo di messaggeri il cui lavoro è proprio quello di recapitare i messaggi.

Il solo modo di spedire messaggi è scriverli su una grande tavoletta di pietra che i messaggeri recapitano a destinazione. Le tavolette di pietra sono di una dimensione fissa e hanno soli 6 spazi in cui scrivere, Ogni spazio può contenere una lettera o un numero. I messaggi sono spesso suddivisi su un certo numero di tavolette, ma poiché le tavolette sono molto pesanti (sono di pietra) solo una alla volta può essere recapitata.

I messaggeri non sono tipi di cui fidarsi molto, poiché sono pigri e spesso scordano quello che devono fare. Non bastasse, spesso si fermano per lunghe pause durante l'orario di lavoro e di tanto in tanto cercano di fuggire dalla città.

I Governanti vogliono così trovare una maniera per comunicare in modo affidabile, cercando di sviluppare un insieme di regole condivise. L'applicazione di queste regole permette loro di conoscere se i loro messaggi sono stati recapitati e se il messaggio era corretto (quindi non manomesso lungo il percorso). I Governanti hanno già deciso che la destinazione dovrebbe essere scritta sulla tavoletta.

Nei gruppi lo scopo è quello di sviluppare le regole che i Governanti devono mettere in atto per comunicare...

Cosa c'entra tutto questo?

In internet i dati sono suddivisi in pacchetti per essere trasmessi. I canali di comunicazione che li trasportano non sono sempre affidabili. Singoli pacchetti, di tanto in tanto, vengono danneggiati, perduti o mescolati rispetto all'ordine che avevano originariamente.

Nelle Tavolette di Pietra, le tavolette sono i pacchetti e il loro contenuto sono i dati. I pacchetti contengono sia dati sia informazioni legate al protocollo (header). La dimensione dell'header determina quanti dati possono essere trasferiti, così è necessario trovare un bilanciamento tra i dati e le informazioni di header, poiché i pacchetti sono di lunghezza finita.

Gli studenti scopriranno che avranno necessità di scambiare alcuni dei dati presenti nelle tavolette, come il numero di pacchetto e il numero totale di pacchetti spediti, oltre a verificare se il pacchetto rappresenta una conferma di ricezione. Poiché queste azioni aggiungono dati, nel complesso saranno necessari ulteriori pacchetti rispetto a quelli pensati in origine.

I protocolli internet come TCP e UDP in generale bilanciano tutti i fattori con l'obiettivo di creare comunicazioni affidabili ed efficienti.

Questa attività è riadattata da quella presentata nel progetto "Computing Science Inside" (csi.dcs.gla.ac.uk)

Parte III

**Dire ai computer cosa devono
fare — *Rappresentare le
procedure***

Dire ai computer cosa devono fare

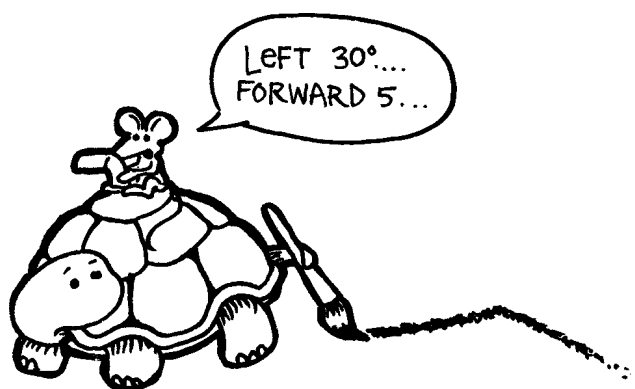
I computer seguono le istruzioni, milioni di istruzioni per ogni secondo. Per dire ad un computer cosa deve fare tutto ciò che serve è dare le istruzioni giuste. Ma non è così semplice come sembra!

Quando ci vengono date istruzioni da seguire noi usiamo il buon senso e l'esperienza per interpretare quale possa essere il significato. Se qualcuno dicesse "passa attraverso quella porta", chiaramente non intenderebbe che dobbiamo abbattere la porta ma che dobbiamo oltrepassare la porta ed è quindi necessario prima aprire la porta se questa è chiusa! I computer funzionano diversamente. Per esempio quando un computer controlla un robot occorre prendere le dovute precauzioni perché non possa causare danni e pericolo interpretando letteralmente le istruzioni, come l'ordine di passare attraverso una porta. Occorre un po' di tempo e di esperienza perché ci si possa abituare di avere a che fare con qualcosa che segue sempre le istruzioni alla lettera, senza "pensare".

Le due attività di questa parte del testo servono per fornire alcune idee su come si possa comunicare con una macchina che obbedisce alla lettera usando un insieme prefissato di istruzioni.

La prima attività mostra il funzionamento di una "macchina" che i computer usano per riconoscere parole, numeri e stringhe (sequenze) di caratteri. Queste "macchine" sono chiamate "automi a stati finiti".

La seconda attività ci introduce alla comunicazione con i computer. Un buon programmatore deve imparare come parlare ai computer usando un prefissato insieme di istruzioni che verranno interpretate letteralmente. La sequenza delle istruzioni è il programma. Ci sono molti diversi linguaggi di programmazione fra i quali un programmatore può scegliere per scrivere le istruzioni, ma noi useremo un semplice linguaggio che si può scrivere senza un computer.



(a sinistra 30 gradi, avanti di 5...)

Attività 12

Caccia al tesoro — *Automati a stati finiti*

Sommario

I programmi per computer spesso hanno necessità di elaborare sequenze di simboli come lettere o parole in un documento o anche il testo di un altro programma. Gli scienziati informatici spesso usano un “automa a stati finiti” per questi compiti. Un automa a stati finiti (Finite State Automata o FSA in inglese) segue un insieme di istruzioni per capire se il computer deve riconoscere la parola o la sequenza di simboli. Noi lavoreremo con qualcosa di equivalente a un automa a stati finiti: la mappa di un tesoro!

Competenze richieste

- ✓ Lettura di semplici mappe
- ✓ Riconoscimento di campioni (pattern)
- ✓ Logica
- ✓ Capacità di seguire le istruzioni

Età

- ✓ A partire da 9 anni

Materiale

L'insegnante deve avere:

- ✓ Un set di carte delle isole (le istruzioni vanno mantenute nascoste da chi tenta di disegnare la mappa!)

Fotocopiate le pagine a partire da 142. Le carte vanno ritagliate, piegate lungo la linea tratteggiata e incollate in modo che da un lato sia presente il nome dell'isola e sull'altro le istruzioni.

Ogni studente deve avere:

- ✓ Il foglio di lavoro: Trova la via per le ricchezze dell'Isola del Tesoro (pagina 141)
- ✓ Penna o Matita

Ci sono anche attività opzionali di approfondimento, per le quali ogni studente deve avere:

- ✓ Il foglio di lavoro: l'Isola del Tesoro (pagina 147)

✓ Il foglio di lavoro: Il misterioso gioco della moneta (pagina 149)

L'Isola del Tesoro

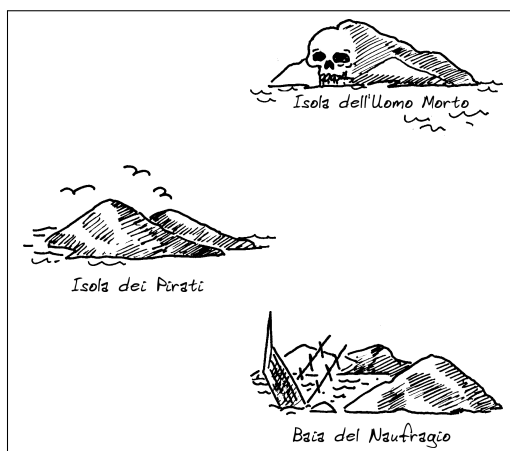
Introduzione

Il vostro scopo è di trovare l'Isola del Tesoro. Navi pirata amiche veleggiano lungo un insieme di rotte fra le isole di questa parte del mondo, offrendo passaggi ai viaggiatori. Ogni isola ha due navi in partenza, A e B, sulle quali potete viaggiare. Dovete trovare la migliore via per l'Isola del Tesoro. Una volta giunti su di un'isola potete chiedere di ripartire o con la nave A o con la nave B (non entrambe). La persona all'isola vi dirà quindi su quale isola vi porterà la nave che avete scelto, ma i pirati non hanno una mappa con tutte le isole e le rotte. Dovete usare la vostra mappa per tenere traccia man mano di dove siete diretti e delle isole sulle quali siete già stati.

Esempio pratico

(Nota: La mappa utilizzata è diversa da quella che gli studenti useranno nell'attività)

Usando un proiettore o la lavagna, mostrate o disegnate il diagramma delle tre isole come nella figura che segue:

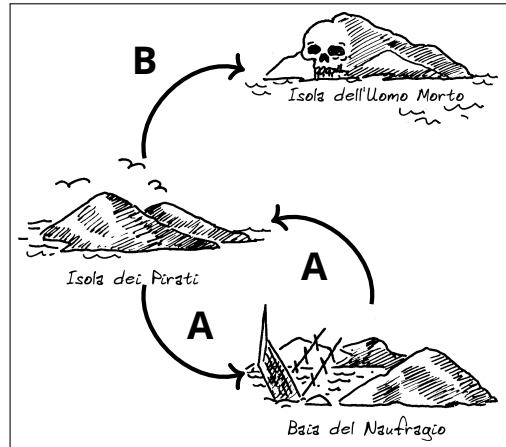


Copiate, ritagliate e piegate le tre carte presenti nelle prossime due pagine. Scegliete tre studenti e fornite loro le carte in modo che ogni studente possa leggere solamente le regole della propria isola e che il resto della classe veda solo la corrispondenza fra i tre studenti e le tre isole. NB: I percorsi indicati in questo esempio saranno diversi da quelli dell'attività.

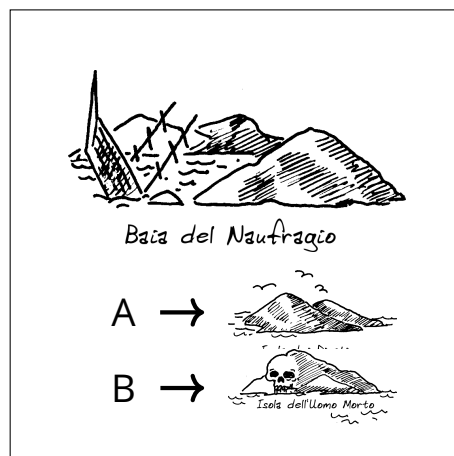
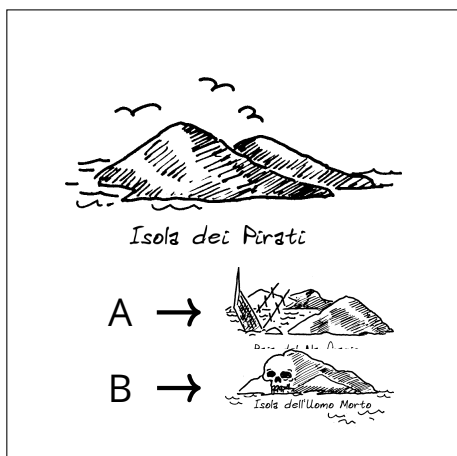
Partendo dall'Isola dei Pirati chiedete di usare la nave A. Lo studente con la carta dell'Isola dei Pirati vi dirà che la nave porta alla Baia del Naufragio. Segnate questo percorso sulla mappa. Alla Baia del Naufragio chiedete ancora di usare la nave A. Lo studente con la carta della Baia del Naufragio vi dirà che la nave vi riporta indietro all'Isola dei Pirati. Segnate anche questo percorso sulla mappa. Chiedete allora per

la nave B che vi condurrà all'Isola dell'Uomo Morto. Segnate anche questo sulla mappa. A questo punto siete bloccati.

La vostra mappa alla fine dovrebbe apparire come questa:



Carte per l'esempio pratico



Carte per l'esempio pratico



Attività

Scegliete sette studenti che saranno le “isole”. Ognuno di questi sette studenti terrà la carta che rappresenta la propria isola, mantenendo le istruzioni segrete verso di sé in modo che il resto della classe non possa vederle. Posizionate questi studenti in modo casuale nella classe o nel giardino in modo che non siano troppo vicini fra loro. A tutti gli altri studenti date una copia della mappa bianca, con solo il disegno delle isole. Lo scopo dell'attività è di navigare per trovare una rotta che giunga dall'Isola dei Pirati all'Isola del Tesoro, indicandola sulla mappa a loro disposizione come spiegato nell'esempio pratico (è consigliabile che gli studenti facciano il percorso uno alla volta perché non sentano in anticipo le rotte dei compagni).

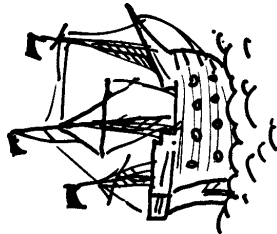
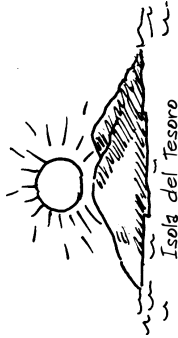
Per chi finisce velocemente: cercare altre rotte alternative.

La mappa completa dovrebbe apparire così:

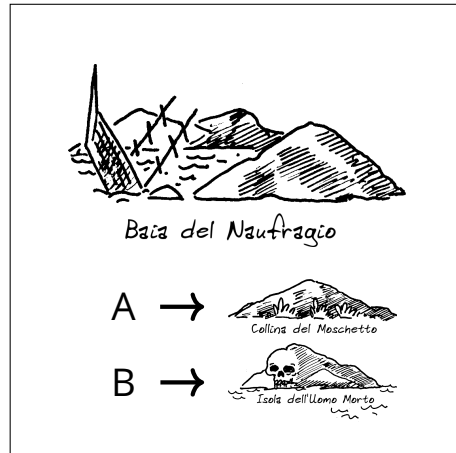
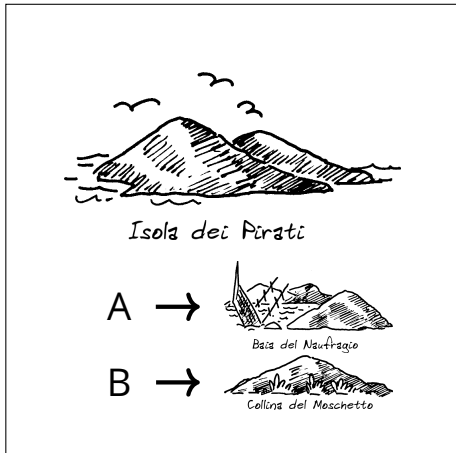
Discussione

Qual è la rotta più veloce? Quale sarebbe una rotta molto lenta. Alcune rotte possono contenere cicli. Potete trovare esempi di rotte con cicli? (Per esempio, BBBABAB e BBBABBABAB, tutte portano all'Isola del Tesoro).

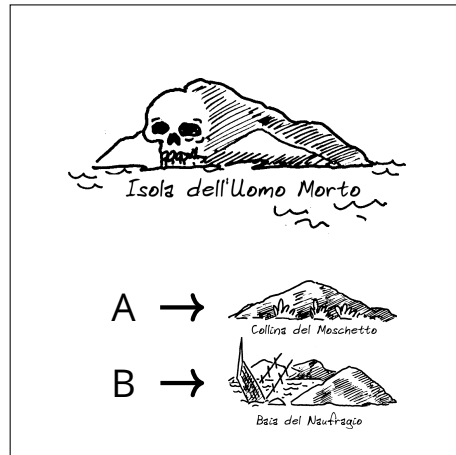
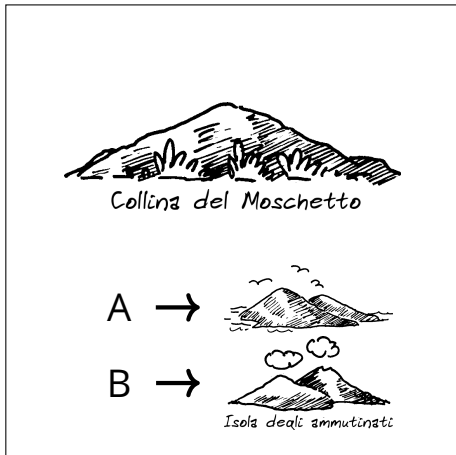
Foglio di Lavoro: Trova la via per le ricchezze dell'Isola del Tesoro



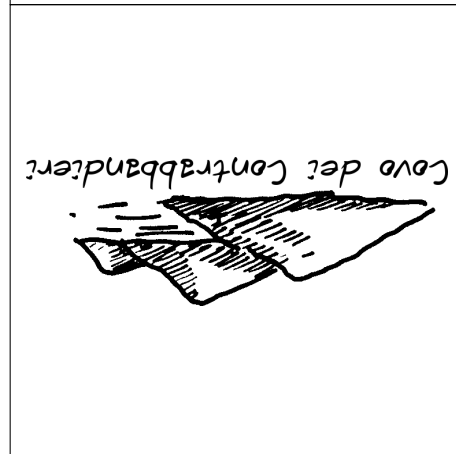
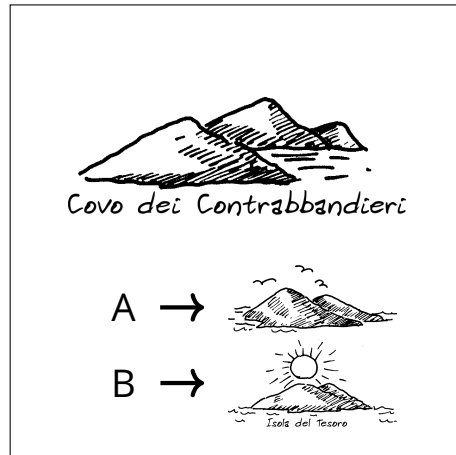
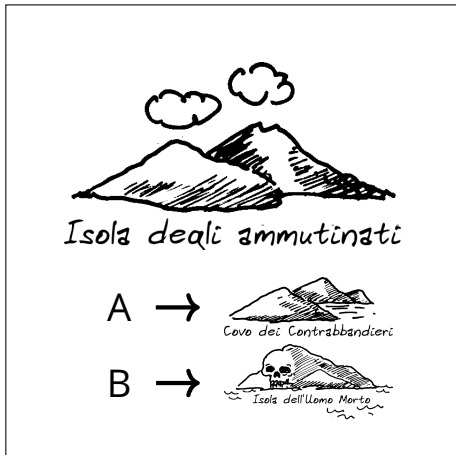
Pagina da fotocopiare: carte delle isole (1/4)



Pagina da fotocopiare: carte delle isole (2/4)



Pagina da fotocopiare: carte delle isole (3/4)

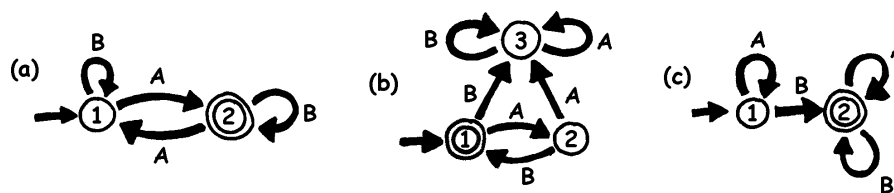


Pagina da fotocopiare: carte delle isole (4/4)



Automati a stati finiti

Un altro modo di disegnare una mappa è il seguente:



Le isole sono rappresentate da cerchietti con indicato un numero, l'isola di arrivo ha un doppio cerchietto. Quali rotte dobbiamo attraversare per arrivare all'isola finale negli esempi della figura qui sopra?

Soluzioni

Nella mappa (a) si arriva all'isola finale (la numero 2) solo se la sequenza contiene un numero dispari di A (per esempio AB, BABAA o AAABABA)

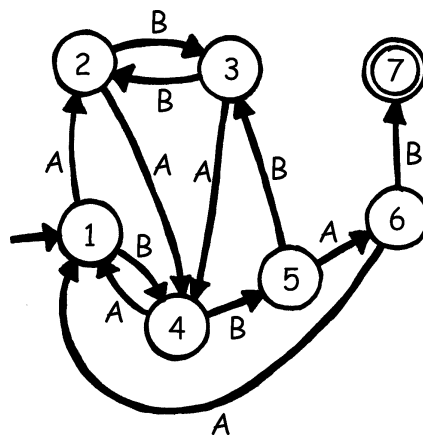
Nella mappa (b) si arriva all'isola finale solo con sequenze (anche vuote) di A e B dove i sue simboli si alternano fra loro (AB, ABAB, ABABAB, ...).

La mappa (c) richiede che la sequenza contenga almeno una B (le sole sequenze non accettate sono A, AA, AAA, AAAA, ...).

Foglio di lavoro: l'isola del tesoro

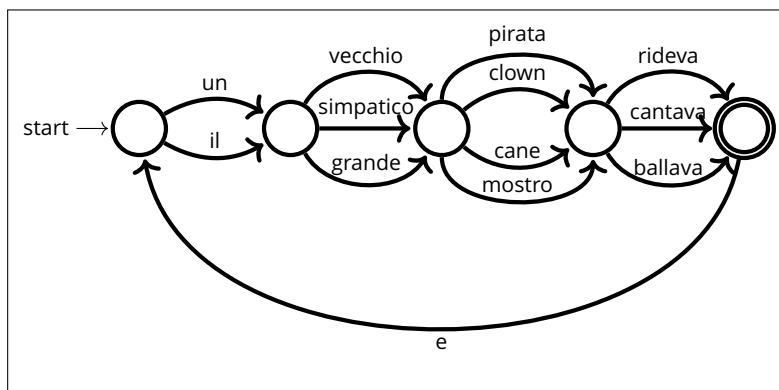
Siete capaci di sotterrare bene il vostro tesoro? Quanto potete rendere difficile trovare il tesoro? È ora di creare la vostra mappa!

1. Ecco una versione diversa della stessa idea di rappresentare una mappa. Questa è la mappa del precedente esercizio. Gli scienziati informatici usano questo metodo semplice e facile per disegnare le mappe utili a cercare particolari sequenze di simboli (dette pattern).



Disegnate la vostra mappa (simile a quella qui sopra ma con percorsi differenti) così da poter vedere chiaramente le rotte sulle quali viaggeranno le vostre navi pirata. Create quindi la vostra mappa bianca e le vostre carte delle isole. Qual è il percorso più efficiente per raggiungere la vostra isola del tesoro?

2. I vostri amici riescono a seguire bene la vostra mappa? Date loro una sequenza di A e di B e vedete se raggiungono l'isola corretta. Si possono fare tanti giochi e puzzle basati su questa idea di automa a stati finiti.
3. Ecco un modo per costruire frasi scegliendo casualmente percorsi attraverso la mappa che segue e trascrivendo le parole incontrate.



Ora provate a fare uno schema simile, magari potete anche costruire una storia divertente!

Foglio di lavoro: il misterioso gioco della moneta.

Alcuni amici hanno scaricato da Internet un gioco nel quale un robot lancia in aria una moneta e occorre indovinare se viene testa o croce. All'inizio il gioco sembra facile, sembrano esserci 50 possibilità su 100 di vincere o così pensano i ragazzi! Dopo un po' iniziano ad essere sospettosi. Sembra esserci una sequenza, un pattern ripetuto nel modo in cui la moneta appare. È un gioco truccato? Sicuramente no, pensano. Ma decidono ugualmente di investigare e Giovanni scrive i risultati di tutti i tiri della moneta (t=testa, c=croce):

```
t t c t t c t t t c c t t t t c c t c c c t t t t t c t t t c c
c t t t c c c t t t t t t c c t c c c c t c c t c c c t t t c
c t t t c t t t t t t t t t c c t t t c c c c t t t t t c c c c
c c c
```

I nostri amici si devono ricredere, la sequenza non sembra proprio casuale. Potete trovare anche voi un pattern per prevedere i risultati dei prossimi tiri? C'è una semplice mappa che descrive le sequenze dei tiri, provate a vedere se riuscite a trovarla (suggerimento: è composta da 4 isole).

Cosa c'entra tutto questo?

Gli automi a stati finiti sono usati in informatica per riconoscere sequenze di caratteri o di eventi.

Un semplice esempio lo potete vedere, o meglio ascoltare, quando telefonando a un'azienda o a un call center vi accoglie un messaggio registrato che dice "premi 1 per questo... premi 2 per quello... o premi 3 per parlare con un operatore umano". Il tasto del telefono è l'input per una macchina a stati finiti. Il percorso nella macchina a stati finiti telefonica può essere semplice o complesso. Talvolta potete anche ritornare a un menù già visitato perché c'è un ciclo nell'automa a stati finiti. Quando questo accade è un errore di progetto del sistema ed è molto frustrante per chi chiama!

Un altro esempio lo potete trovare negli sportelli bancomat. Il programma della macchina vi conduce lungo una sequenza di eventi. All'interno di questo programma c'è un automa a stati finiti. Ogni tasto che premete porta l'automa in un nuovo stato (il nome informatico delle "isole"). Alcuni stati compiono azioni quali "fornisci 100 euro" o "stampa lo scontrino" oppure ancora "espelli la tessera".

Alcuni programmi veramente trattano frasi in italiano o in inglese usando mappe come quella di pagina 114. Possono essere usati per generare frasi e per elaborare frasi in ingresso. Negli anni 1960 uno scienziato informatico scrisse un programma famoso chiamato "Eliza" (dal nome di Eliza Dolittle, la protagonista di *My Fair Lady*) che era in grado di sostenere conversazioni con le persone. Il programma pretendeva di essere uno psicologo e faceva domande come "Mi dica qualcosa della sua famiglia" e anche "Suvvia, vada avanti". Anche se il programma non "capiva" nulla era sufficientemente credibile e alcuni utenti umani erano veramente convinti di parlare con un reale psicologo.

Anche se i computer non sono bravi a comprendere il linguaggio naturale, possono elaborare rapidamente linguaggi artificiali. Un tipo importante di linguaggio artificiale sono i linguaggi di programmazione. I computer usano macchine a stati finiti per leggere i programmi e trasformarli nella forma di istruzioni elementari del computer, che possono essere "eseguite" direttamente dal computer stesso.

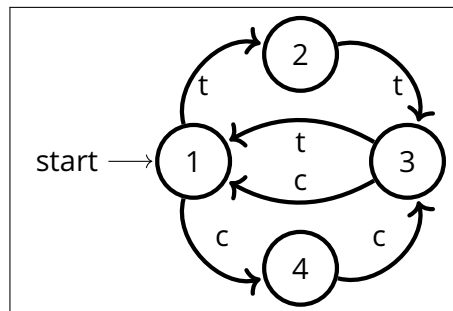


(Grr! Questa non può essere una mappa del tesoro! Dove è la X?)

Soluzioni e suggerimenti

Il misterioso gioco della moneta (pagina 149)

Il misterioso gioco della moneta usa questa mappa per i risultati dei lanci delle monete:



Se lo seguirete, scoprirete che i primi due tiri ogni tre danno sempre lo stesso risultato.

Attività 13

Gli ordini di marcia — *I linguaggi di programmazione*

Sommario

I computer vengono programmati mediante un “linguaggio” composto da un numero limitato di ordini che possono essere eseguiti. Uno dei lati più frustranti dello scrivere programmi è che i computer obbediscono sempre alla lettera agli ordini impartiti, anche quando per colpa di “malintesi” questi producono un risultato chiaramente diverso da quello atteso. Questa attività fornisce agli studenti alcune esperienze su questo aspetto della programmazione.

Competenze richieste

- ✓ Saper dare e seguire sequenze di istruzioni

Età

- ✓ A partire da 7 anni

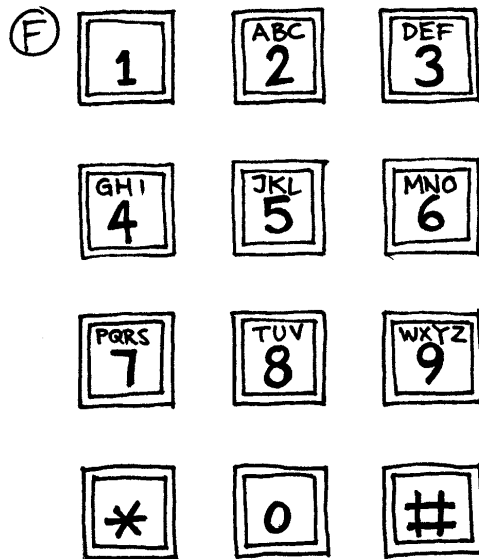
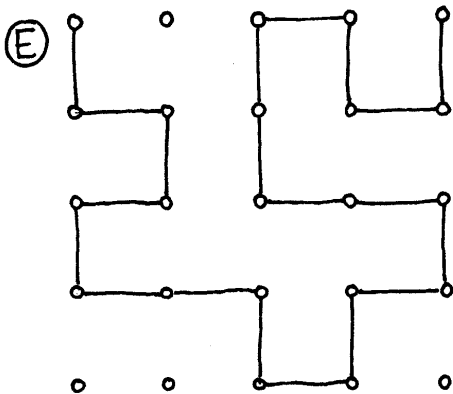
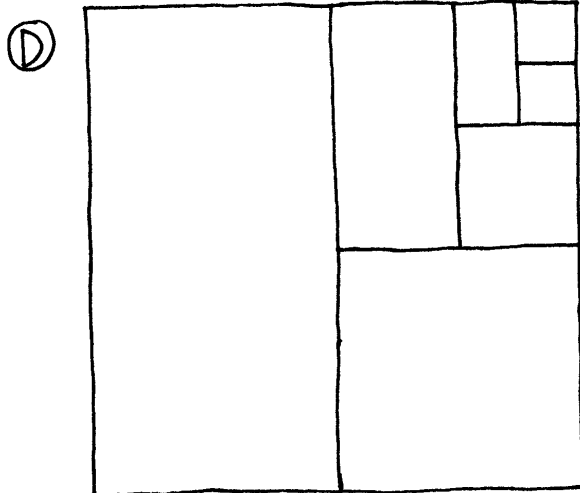
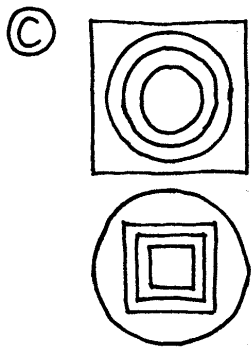
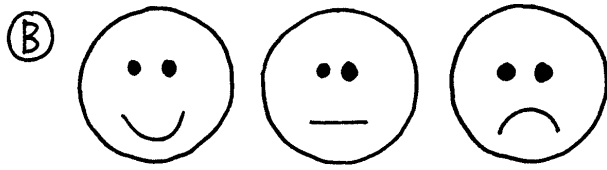
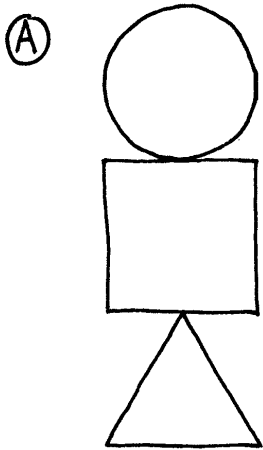
Materiale

Il docente deve avere:

- ✓ Carte con figure come quelle mostrate nella pagina seguente.

Ogni studente deve avere:

- ✓ Matita, carta e righello.



Ordini di marcia

Introduzione

Discutere se sarebbe bene che tutte le persone seguissero esattamente le istruzioni. Per esempio cosa succederebbe se indicando una porta chiusa qualcuno desse l'ordine "Passa attraverso quella porta"?

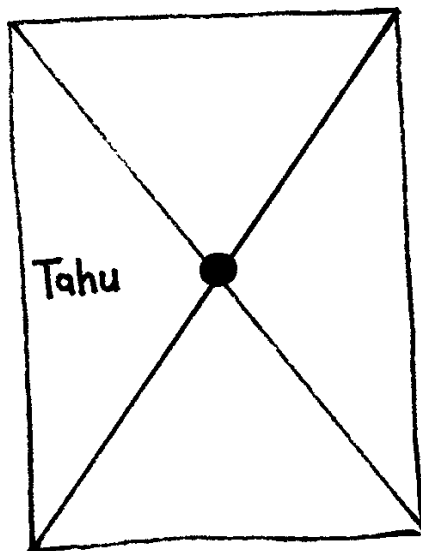
I computer lavorano seguendo una lista di istruzioni e fanno esattamente quello che le istruzioni dicono anche quando le istruzioni non hanno alcun senso!

Esempio pratico

Chiedete agli studenti di disegnare su un foglio bianco una figura seguendo attentamente le istruzioni seguenti:

1. Disegna un punto nel centro del foglio.
2. A partire dall'angolo in alto a sinistra traccia una riga dritta che passi attraverso il punto disegnato e termini all'angolo in basso a destra.
3. A partire dall'angolo in basso a sinistra traccia una riga dritta che passi attraverso il punto disegnato e termini all'angolo in alto a destra.
4. Scrivi ora il tuo nome nel triangolo posto al centro del lato sinistro del foglio.

Il risultato dovrebbe apparire simile a questo:



Attività

Scegliete uno studente e dategli una immagine (per esempio uno degli esempi di pagina 154). L'immagine deve rimanere nascosta al resto della classe. Lo studente deve descrivere l'immagine al resto della classe, che deve tentare di riprodurla seguendo esclusivamente le istruzioni impartite. Gli altri studenti possono chiedere di precisare le istruzioni qualora non risultassero chiare. L'obiettivo dell'esperimento è di vedere quanto velocemente e accuratamente l'esercizio venga completato.

Ripetete l'esperimento, questa volta non consentendo più domande da parte della classe. È meglio scegliere immagini semplici perché gli studenti molto facilmente potranno perdersi.

Ora ripetete l'esperimento con lo studente che fornisce le informazioni nascosto dietro uno schermo in modo che non ci sia alcuna forma di comunicazione se non la sequenza delle istruzioni.

Fate notare che è proprio questa la forma di comunicazione che i programmatori hanno con il computer quando scrivono programmi. Essi infatti scrivono sequenze di istruzioni per il computer e non possono vedere l'effetto se non più tardi, quando il programma verrà eseguito.

Ora potete far fare semplici disegni agli studenti e chiedere che scrivano la sequenza delle istruzioni usate per realizzare il disegno. Possono fare l'esercizio a coppie (scambiandosi le istruzioni per vedere di ottenere nuovamente il disegno del collega) oppure uno studente alla volta può leggere le sue istruzioni al resto della classe e confrontare i disegni generati da tutti i colleghi.

Variazioni

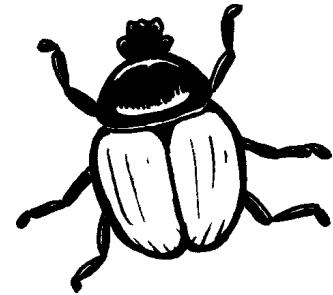
1. Scrivere le istruzioni per realizzare una freccetta o un aeroplanino di carta.
2. Scrivere le istruzioni per raggiungere un punto misterioso nella scuola o nel cortile con indicazioni come "avanza di x metri", "gira a sinistra" (90 gradi), "gira a destra" (90 gradi). Gli studenti dovrebbero provare le istruzioni più e più volte fino a riuscire ad ottenere il risultato voluto.
3. Mosca cieca. Bendate uno studente e chiedete agli altri di dargli le istruzioni per muoversi nella classe e compiere alcune azioni.

Cosa c'entra tutto questo?

I computer funzionano seguendo una lista di istruzioni chiamato *programma*. Ogni programma è stato scritto per far svolgere a un computer un particolare compito. I programmi sono scritti con linguaggi che creati appositamente, costituiti un insieme limitato di istruzioni, che servono a dire al computer cosa fare. Alcuni linguaggi sono più adatti ad alcuni compiti rispetto ad altri.

Indipendentemente dal linguaggio di programmazione usato, i programmatori devono stare molto attenti a specificare esattamente cosa vogliono che il computer faccia. A differenza dagli esseri umani, un computer esegue le istruzioni alla lettera anche se queste sono palesemente errate, magari ridicole.

È importante allora che i programmi siano scritti bene. Un piccolo errore può causare molti problemi. Immaginate le conseguenze che l'errore di un programma può avere nel computer che controlla il lancio di un missile di una missione spaziale o nel computer di controllo di una centrale nucleare o in quello che accende i segnali di controllo per i treni! Gli errori dei programmi sono di solito chiamati "bug" cioè scarafaggi in onore (si fa



per dire) di un "bug" che fu rimosso da un relais elettromagnetico di uno dei primi computer all'inizio degli anni 1940. Lo scarafaggio aveva causato il malfunzionamento del computer e ancora oggi la ricerca degli errori nei programmi viene chiamata "debugging" cioè disinfestazione.

Più un programma è complesso, più errori può contenere. Negli USA la possibile presenza di errori nei programmi è diventato un problema enorme nel progetto del Ministero della Difesa chiamato "Guerre Stellari". Il progetto prevede la creazione di un sistema controllato da computer che crei uno scudo impenetrabile per difendersi da attacchi nucleari. Alcuni scienziati informatici hanno sostenuto che il progetto potrebbe non funzionare per la complessità e quindi la conseguente inaffidabilità dei programmi necessari. I programmi devono essere controllati accuratamente per cercare più bug che si può, ma non si può avere alcuna certezza di averli trovati tutti. E non è certo il caso di sparare missili nucleari per vedere se funziona davvero oppure no!

Parte IV

Problemi veramente difficili — *Intrattabilità*

Problemi veramente difficili

Esistono problemi che sono troppo difficili anche per un computer? Sì! Vedremo nell'attività 21 che anche solo sostenere una conversazione o una chiacchierata sono attività che i computer non possono fare, non perché non possano parlare ma perché non possono comprendere o pensare a qualcosa di sensato da dire. Noi umani non sappiamo come pensiamo e come elaboriamo le domande in una conversazione e quindi non possiamo dire ad un computer come fare. Ma non è questo il tipo di difficoltà che verrà trattato in questa sezione del testo. Qui daremo un'occhiata a problemi la cui soluzione può essere descritta facilmente tramite un programma ma il computer non può risolverli perché occorrerebbe troppo tempo: forse anche milioni di secoli. Non conta inventare o acquistare computer più veloci e potenti: se anche riuscissimo a costruire o a comperare computer cento volte più potenti occorrerebbero sempre milioni di anni e se fossero un milione di volte più potenti servirebbe sempre un secolo. Questo è ciò che chiamiamo un problema difficile o intrattabile: un problema per il quale anche il più veloce elaboratore impiegherebbe più della vita di un essere umano per dare una risposta.

Le attività della parte II sugli algoritmi hanno mostrato come trovare il modo di rendere i programmi più efficienti. In questa sezione vedremo problemi per i quali non sono conosciute soluzioni efficienti, problemi che possono richiedere milioni di secoli di elaborazione per esser risolti. E incontreremo il mistero più appassionante dell'informatica di oggi. Potrebbe essere che nessuno sia stato in grado finora di trovare una soluzione oppure la soluzione efficiente potrebbe semplicemente non esistere. Nessuno lo sa. Ma c'è di più. Ci sono migliaia di problemi che, sebbene appaiano differenti, sono in realtà equivalenti, nel senso che se un metodo efficiente fosse trovato per risolvere uno di essi, la soluzione potrebbe essere convertita in un metodo efficiente per risolvere tutti gli altri. Nelle attività di questa sezione giocheremo con questi problemi.

Per gli insegnanti

Ci sono tre attività in questa sezione del testo. La prima è relativa alla colorazione delle mappe, consiste nel trovare il numero minimo di colori necessari per colorare sempre i paesi confinanti con colori differenti. La seconda richiede la capacità di usare una semplice mappa stradale e di collocare camion di gelati agli incroci in modo che nessuno debba andare troppo lontano per mangiare un gelato. La terza è un'attività da svolgere all'aria aperta: usando fili e mollette da bucato occorre trovare la rete di lunghezza minima in grado di connettere un insieme di punti.

Queste attività forniscono una prova tangibile dell'idea di complessità, di come problemi che sono semplici da definire possano trasformarsi in problemi incredibilmente difficili da risolvere. Questi problemi non sono

astrusi, ma al contrario sono presenti in tante questioni pratiche che fanno parte della vita di tutti i giorni, come disegnare una mappa, creare l'orario delle lezioni, progettare il percorso per la costruzione di strade. La definizione formale del problema trattato in questa sezione ricade sotto la nozione di "NP-completezza" come viene spiegato nelle sezioni *Cosa c'entra tutto questo?* presenti alla fine di ogni attività. Anche se le attività stesse possono essere svolte in qualsiasi ordine, i capitoli dovrebbero essere letti nell'ordine nel quale vengono presentati. Il lettore, una volta giunto al termine, avrà afferrato completamente la più importante questione ancora aperta dell'Informatica moderna.

Il nome tecnico di questa sezione è "intrattabilità": questi problemi, che sono difficili da risolvere, vengono detti intrattabili. La parola viene dal latino *tractare* che significa maneggiare, trascinare. Trattabile assume oggi il significato di "gestibile". I problemi intrattabili sono quelli che non sono facilmente gestibili perché il tempo necessario per avere una risposta sarebbe troppo lungo. Anche se potrebbe sembrare una questione esoterica, l'intrattabilità è un tema di grande interesse pratico perché una conquista in questo campo avrebbe immediate ricadute in tantissime aree di ricerca. Per esempio, molti codici crittografici sono basati sulla intrattabilità di qualche problema. Un criminale che fosse capace di trovare una soluzione efficiente ad un problema intrattabile potrebbe avere mano libera nella decodifica dei segreti. Potrebbe vendere informazioni o, più semplicemente, creare falsi conti bancari. Studieremo questi problemi nella parte V: Crittografia.

Attività 14

Il cartografo povero — *La colorazione dei grafi*

Sommario

Molti problemi di ottimizzazione sono relativi a situazioni nelle quali certi eventi non possono accadere nello stesso tempo o nelle quali i componenti di un insieme di oggetti non possono essere adiacenti. Per esempio, chiunque abbia provato a calcolare l'orario delle lezioni della scuola ha dovuto affrontare il problema di soddisfare i vincoli richiesti da tutte le persone coinvolte. Molte di queste difficoltà sono presenti nel problema di colorazione delle carte geografiche nelle quali il colore di ogni stato deve essere scelto in modo tale che gli stati limitrofi abbiano sempre colori distinti. Il problema della colorazione delle carte geografiche viene affrontato in questa attività.

Competenze richieste

- ✓ Capacità di risolvere problemi
- ✓ Capacità di ragionamento logico
- ✓ Conoscenza delle procedure algoritmiche e della complessità
- ✓ Capacità di comunicazione

Età

- ✓ A partire dai 7 anni

Materiale

- ✓ una lavagna (o una simile superficie scrivibile)

Ogni studente deve avere:

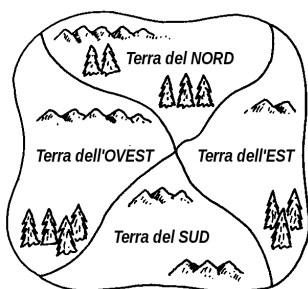
- ✓ copia di uno o più fogli di lavoro
- ✓ piccoli indicatori colorati (per esempio piccoli gettoni)
- ✓ quattro pennarelli/matite di differenti colori (corrispondenti ai colori dei gettoni)

La colorazione dei grafi



Introduzione

Questa attività ruota attorno ad una storia nella quale gli studenti sono chiamati ad aiutare un cartografo, cioè un disegnatore di mappe, che deve colorare gli stati su una carta geografica. Non importa quale colore venga assegnato ad ogni stato, a patto che sia differente dal colore assegnato a tutti gli stati confinanti.



Per esempio questa mappa qui a fianco ha quattro stati. Se coloriamo la Terra del Nord in rosso allora le Terre dell'Est e dell'Ovest non possono essere rosse. Possiamo per esempio colorare la Terra dell'Ovest di verde ed è accettabile che anche la Terra dell'Est sia colorata di verde perché non ha una linea di frontiera con la Terra dell'Ovest (se due stati hanno un solo

punto in comune non si considerano confinanti e possono avere lo stesso colore, non avendo una linea di frontiera che potrebbe generare confusione nella lettura della mappa). La Terra del Sud potrebbe essere colorata di rosso come la Terra del Nord. Quindi questa mappa può essere correttamente colorata usando due soli colori. Nella nostra storia il cartografo è povero e non può permettersi molti pennarelli, quindi l'idea è di consentirgli di usare il minimo numero possibile di colori.

Discussione

Descrivete agli studenti il problema che dovranno risolvere, mostrando il processo di colorazione sulla lavagna.

Consegnate agli studenti il primo foglio di lavoro. Questa mappa può essere correttamente colorata usando due soli colori. Anche se limitare il numero di colori a due soli potrebbe apparire una sfida, questo compito è piuttosto semplice perché le scelte per la colorazione di ogni stato sono quasi automatiche.

Chiedete agli studenti di provare a colorare la mappa con due soli colori. Durante questa fase dovrebbero scoprire la regola del “deve-essere”: una volta che uno stato è stato colorato, tutti quelli confinanti devono venir colorati con l’altro colore. Questa regola viene ripetutamente applicata fino a che tutti gli stati sono stati colorati. È meglio se gli studenti riescono a scoprire questa regola da soli e non meri esecutori della regola dopo che è stata loro spiegata. L’attività di scoprire da soli la regola fornisce loro una migliore percezione del problema: ciò sarà utile nei successivi esercizi.

Quando gli studenti completano questo primo foglio di lavoro si può passare al successivo.

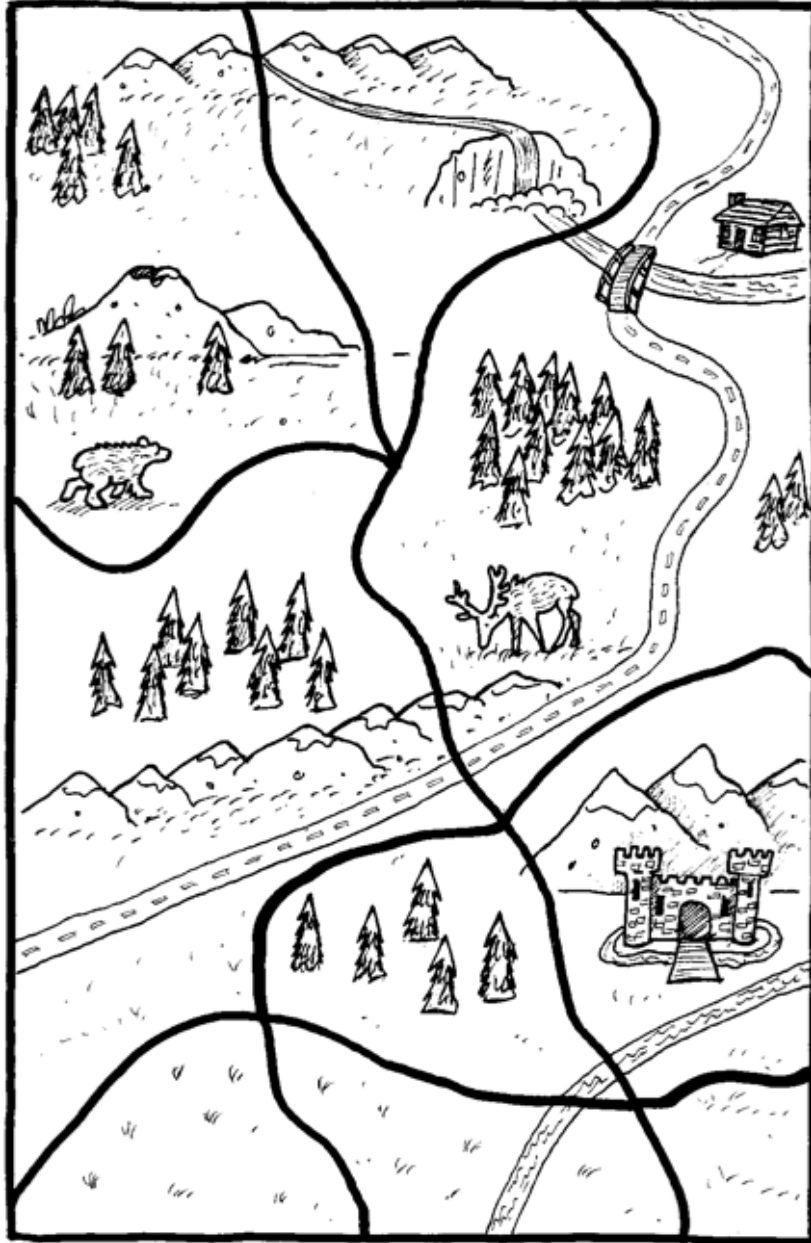
Gli studenti possono scoprire che è meglio usare dei segnaposto, come piccoli gettoni colorati, per provare le soluzioni prima di colorare effettivamente gli stati. Questo consente di rendere più facile provare nuove soluzioni e cambiare idea se le soluzioni dovessero risultare scorrette.

Agli studenti più maturi potete chiedere di spiegare come possono sapere di aver usato il minor numero di colori possibile. Per esempio, almeno tre colori sono necessari nella prima mappa del secondo Foglio di lavoro perché c’è un gruppo di stati (i più grandi) ognuno dei quali confina con gli altri due.

Se uno studente finisce tutti i fogli di lavoro molto in fretta, potete chiedergli di creare una mappa nella quale siano necessari cinque diversi colori. Dato che è stato dimostrato che ogni mappa può essere correttamente colorata con al più quattro colori, questo lavoro terrà impegnato lo studente a lungo! Nella nostra esperienza, gli studenti troveranno facilmente mappe per le quali saranno convinti che siano necessari cinque colori, ma ovviamente sarà sempre possibile trovare una soluzione con soli quattro colori.

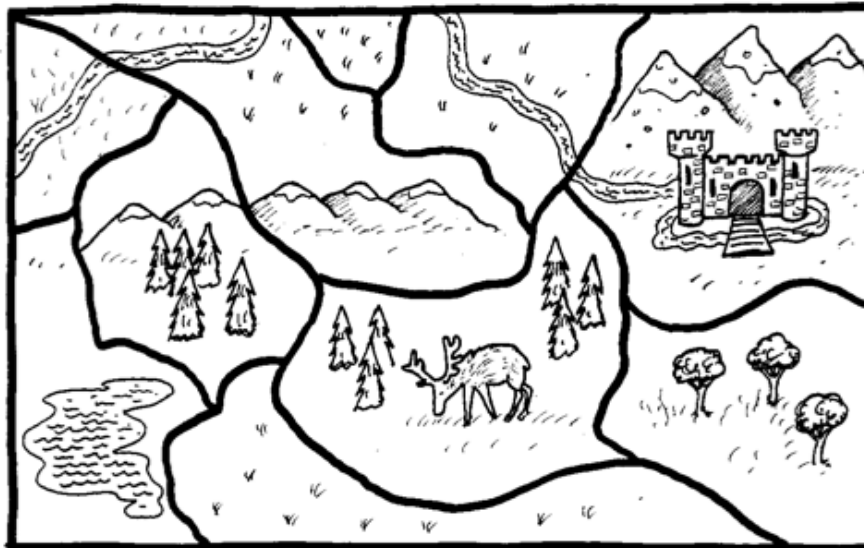
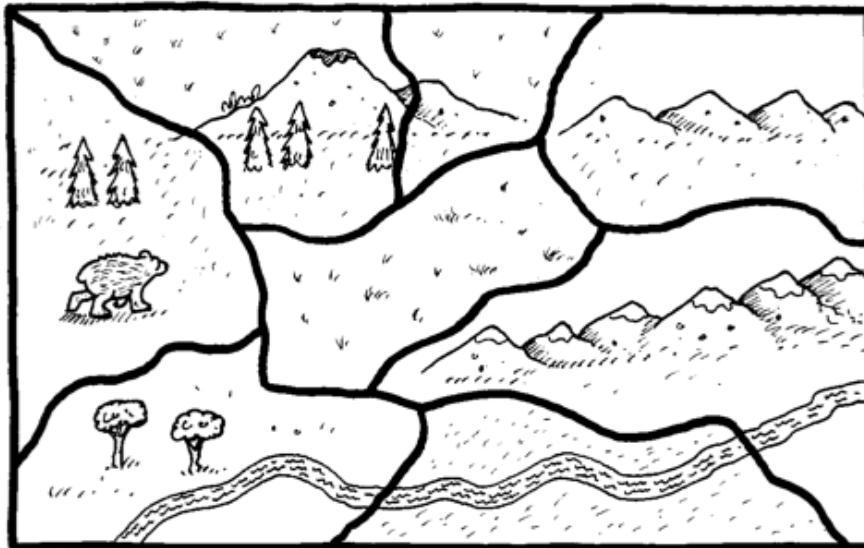
Foglio di Lavoro: Colora la Mappa 1

Colora gli stati di questa mappa usando il minor numero di colori. Fai attenzione che due stati confinanti non siano mai dello stesso colore.



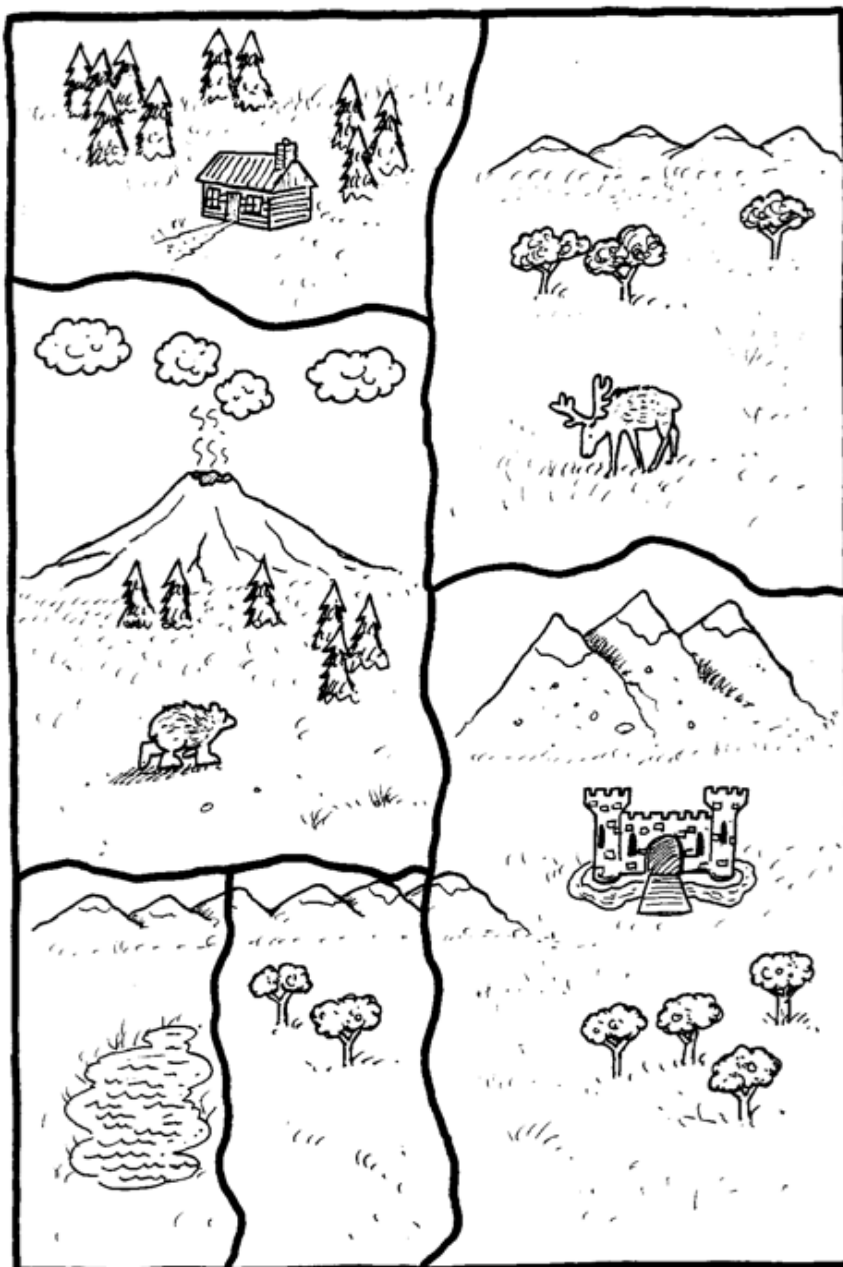
Foglio di Lavoro: Colora la Mappa 2

Colora gli stati delle mappe seguenti usando il minor numero di colori.
Fai attenzione che due stati confinanti non siano mai dello stesso colore.



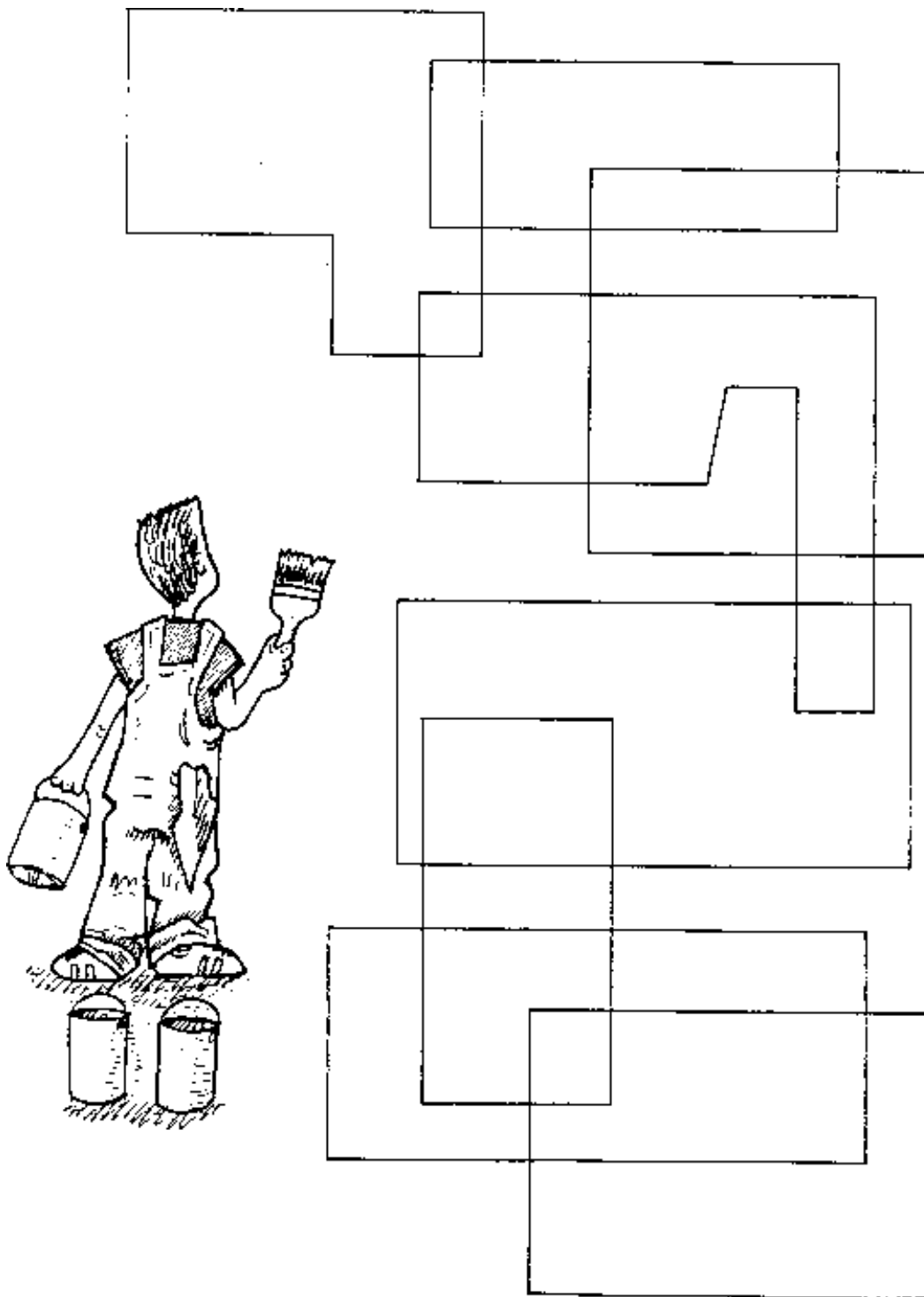
Foglio di Lavoro: Colora la Mappa 3

Colora gli stati di questa mappa usando il minor numero di colori. Fai attenzione che due stati confinanti non siano mai dello stesso colore.



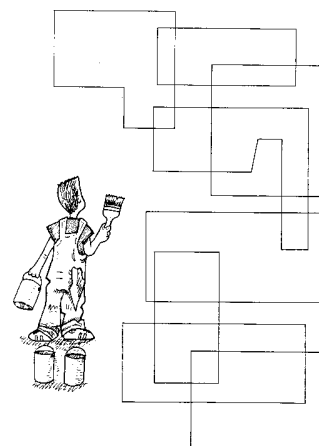
Foglio di Lavoro: Colora la Mappa 4

Colora gli stati di questa mappa usando il minor numero di colori. Fai attenzione che due stati confinanti non siano mai dello stesso colore.



Variazioni ed Estensioni

C'è un modo semplice per costruire mappe che richiedano due soli colori, come mostrato qui. Questa mappa è stata disegnata sovrapponendo curve chiuse (linee tracciate a partire da un punto e che terminano nello stesso punto). Si può disegnare un numero qualsiasi di queste curve, di ogni forma, e la mappa risultante potrà sempre essere colorata con due soli colori.



Quattro colori sono sempre sufficienti a colorare una mappa disegnata su un foglio di carta o su una sfera (come il mappamondo). Uno potrebbe domandarsi (e gli scienziati sono pagati per farlo) quanti colori sono necessari per una mappa disegnata su di un toro (un "toro" per i matematici ha la forma di un salvagente o della camera d'aria di una ruota). In questo caso ci potrebbe essere la necessità di usare cinque colori, ma cinque sono sempre sufficienti. Gli studenti potrebbero divertirsi a ragionare e fare esperimenti per il caso del toro.

Ci sono tante varianti divertenti del problema della colorazione delle mappe che conducono a problemi ancora praticamente irrisolti. Per esempio, se coloro una mappa su di un foglio da solo, so che quattro colori sono sufficienti. Ma supponiamo che invece di lavorare da solo io abbia un collaboratore incompetente (o un avversario), con il quale a turno coloriamo uno stato alla volta. Quanti pennarelli sono necessari per poter, con la mia furbizia, rimediare le mosse poco intelligenti (o addirittura sovversive) del mio collega? Il numero massimo non è conosciuto. Nel 1992 è stato provato che 33 colori sono sempre sufficienti. Nel 2008 questo risultato è stato migliorato dimostrando che 17 dovrebbero essere sufficienti ma ancora non sappiamo quanti siano effettivamente necessari (La congettura degli esperti è che dovrebbero bastare 10 colori, ma non c'è ancora una dimostrazione). Gli studenti possono divertirsi a ricreare questa situazione. Il gioco consiste nella sfida fra due persone nella quale ogni concorrente tenta di massimizzare il numero di colori necessari nella mappa dell'avversario.

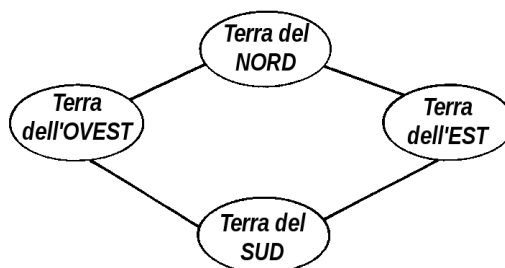
Un'altra versione del problema della colorazione delle mappe è nota come la colorazione dell'impero. Si parte con due mappe distinte aventi lo stesso numero di stati. Ogni stato di una mappa (per esempio sul pianeta Terra) è gemellato con uno stato dell'altra mappa (che potrebbe essere la corrispondente colonia sulla Luna). In aggiunta alla consueta regola che gli stati confinanti non possano avere lo stesso colore, si pone anche la regola che la colonia sulla Luna debba avere lo stesso

colore del corrispondente stato sulla Terra. Quanti colori sono necessari per risolvere questo problema? La risposta è ancora sconosciuta.

Cosa c'entra tutto questo?

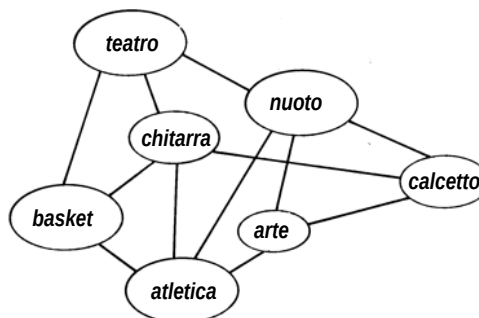
Il problema della colorazione delle mappe che abbiamo discusso in questa attività consiste essenzialmente nel trovare il minimo numero di colori distinti – due, tre o quattro – necessari per colorare una specifica mappa. La congettura che per ogni mappa (piana) fossero sufficienti quattro colori è stata formulata nel 1852, la dimostrazione è stata trovata solo nel 1976. L'informatica è piena di problemi irrisolti. Sapere che la dimostrazione del teorema dei quattro colori ha richiesto 120 anni di attenzione e studio da parte dei ricercatori incoraggia gli studiosi a lavorare sugli altri problemi ancora aperti da decenni.

La colorazione delle mappe appartiene ad una classe più generale di problemi denominata "colorazione dei grafi". Nell'informatica un grafo è una rappresentazione astratta di una relazione, come mostrato qui a lato e come già indicati nella attività 9



(La città fangosa). Il grafo viene disegnato usando cerchi, tecnicamente detti "nodi", per indicare gli oggetti e linee fra essi (detti "archi") per indicare qualche tipo di relazione fra gli oggetti. Il grafo qui sopra corrisponde alla mappa mostrata all'inizio di questa attività. I nodi rappresentano gli stati e gli archi rappresentano la relazione fra due stati che hanno una linea di frontiera in comune. Sul grafo la regola di colorazione dice che due nodi connessi (da un arco) non possono avere lo stesso colore. Contrariamente alle mappe, non esiste limite al numero di colori che un grafo generico può richiedere, perché ogni nodo può essere connesso ad un numero arbitrario di altri nodi, anche a tutti gli altri, mentre la natura bidimensionale delle mappe limita il numero delle possibili configurazioni. Il "problema di colorazione dei grafi" consiste nel trovare il numero minimo di colori necessari per colorare uno specifico grafo.

Nel grafo qui a destra i nodi corrispondono ad attività extrascolastiche svolte dagli studenti di una scuola. Una linea, un arco, fra due attività indica che almeno uno studente desidera svolgere entrambe le attività quindi non dovrebbero essere svolte nello stesso orario. Usando questa rappresentazione, il problema



della colorazione dei grafi consiste nel trovare la tabella oraria che consenta di tenere aperta la scuola per un numero minimo di ore. Gli algoritmi di colorazione dei grafi sono di grande interesse per l'informatica, anche se ormai raramente vengono usati per la colorazione delle mappe! Il nostro cartografo povero è solamente una finzione.

Ci sono letteralmente migliaia di altri problemi basati sui grafi. Alcuni sono descritti in altre parti di questo libro, come il "minimal spanning tree" (attività 9) e il problema degli insiemi dominanti (attività 15). I grafi sono un modo generale di rappresentare i dati e consentono di creare modelli per diverse situazioni come ad esempio mappe stradali con vie e incroci, connessioni fra atomi nelle molecole, percorsi che i messaggi possono percorrere all'interno di una rete di comunicazione fra computer, connessioni fra componenti di un circuito elettronico, relazioni fra le varie fasi richieste per completare un grande progetto. Per questo motivo, i problemi relativi ai grafi hanno grande interesse per gli studiosi di informatica.

Molti di questi problemi sono difficili da risolvere, non difficili concettualmente ma difficili perché richiedono un lungo tempo di elaborazione per ottenere una soluzione. Per esempio, trovare la soluzione più efficiente al problema della colorazione di un grafo di dimensioni non troppo grandi, per esempio al problema del calcolo degli orari delle attività extrascolastiche per trenta docenti e 800 alunni potrebbe richiedere anni, forse secoli di elaborazione anche usando il miglior algoritmo conosciuto e computer potenti. La soluzione potrebbe essere diventata inutile dopo così tanto tempo, sempre assumendo che il computer non si rompa durante tutta questa elaborazione. Tali problemi si risolvono praticamente solo perché ci accontentiamo di ottenere soluzioni molto buone anche se non ottimali. Se insistessimo a volere la soluzione ottimale, questi problemi sarebbero realmente intrattabili.

L'incremento del tempo di elaborazione necessario per calcolare la soluzione del problema di colorazione dei grafi aumenta esponenzialmente rispetto alla dimensione del grafo. Consideriamo il problema di colorazione delle mappe. Può essere risolto provando ogni possibile scelta. Sapendo che bastano quattro colori, occorre provare ogni combinazione di assegnazione dei quattro colori agli stati della mappa. Se gli stati sono n , ci sono 4^n combinazioni possibili. Questo numero cresce molto rapidamente: per ogni stato che viene aggiunto occorre moltiplicare per quattro il numero di casi da esaminare, quindi il tempo necessario per fornire la soluzione. Se, per esempio, avessero inventato un computer capace di risolvere il problema della colorazione dei grafi per cinquanta stati in un'ora e aggiungessimo dieci stati, occorrerebbe più di un anno per trovare la soluzione. Anche il continuo aumento della potenza di elaborazione dei computer non aiuta a risolvere questo tipo di problema.

La colorazione dei grafi è un esempio di problema il cui tempo di soluzione cresce esponenzialmente. Per esempi di piccole dimensioni, come le mappe usate nei fogli di lavoro di questa attività, è abbastanza semplice trovare una soluzione ottima, ma appena il numero degli stati diventa maggiore di dieci, il problema diviene molto difficile da risolvere a mano e con cento o più stati anche un computer può impiegare anni per provare tutti i possibili modi di colorare la mappe e scegliere quello ottimale.

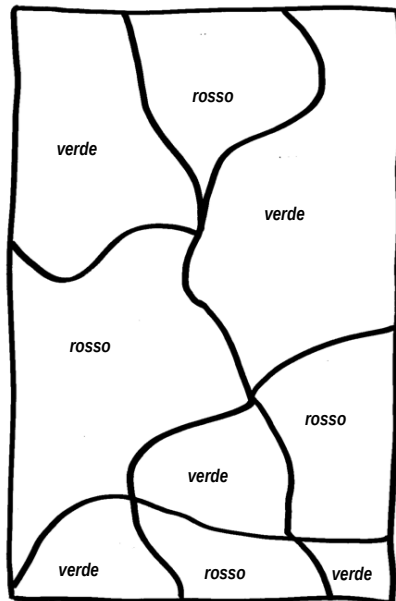
Nella vita reale molti problemi come questi devono essere risolti comunque. Gli informatici usano metodi che sono in grado di fornire soluzioni buone, anche se non perfette. Queste tecniche, dette *euristiche*, sono veloci e forniscono soluzioni spesso molto vicine alle soluzioni ottimali e utilizzabili per i fini pratici. Le scuole possono tollerare di dover usare un'aula aggiuntiva e forse il nostro cartografo può permettersi di acquistare un ulteriore pennarello anche se nella soluzione ottimale potrebbe farne a meno.

Nessuno ha ancora dimostrato che non esiste un modo efficiente di risolvere questo tipo di problema tramite normali computer, ma nessuno ha neanche dimostrato che tale metodo esista. Gli scienziati sono però scettici sulla possibilità che un simile metodo venga trovato. Impareremo di più in merito a questo tipo di problema nelle prossime due attività.

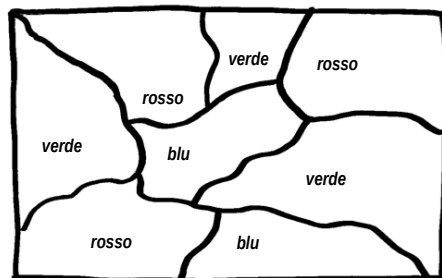
Per ulteriori approfondimenti

Harel discute il teorema dei quattro colori e ne presenta la storia nel libro *Algorithmics* [5], Ulteriori aspetti del problema della colorazione delle mappe sono discussi in *This is MEGA-Mathematics!* di Casey e Fellows [1]. Il libro del 2004 di Kubale dal titolo *Graph Colorings* [9] racconta la storia del problema. Ci sono numerosi siti web relativi a questo argomento.

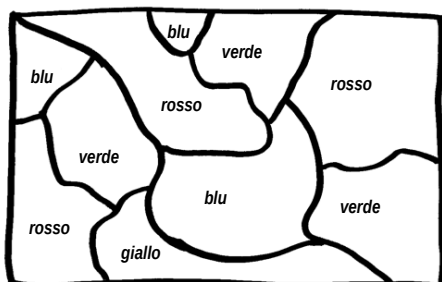
Soluzioni e Consigli

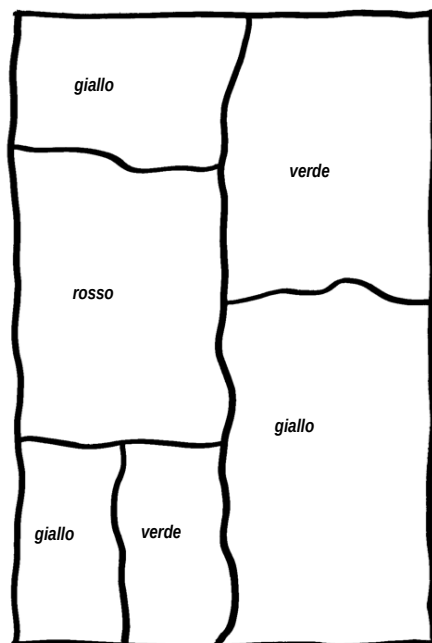


Questa è l'unica soluzione possibile per la mappa del foglio di lavoro 1 (naturalmente la scelta dei colori può essere differente, ma bastano due soli colori per risolvere il problema).

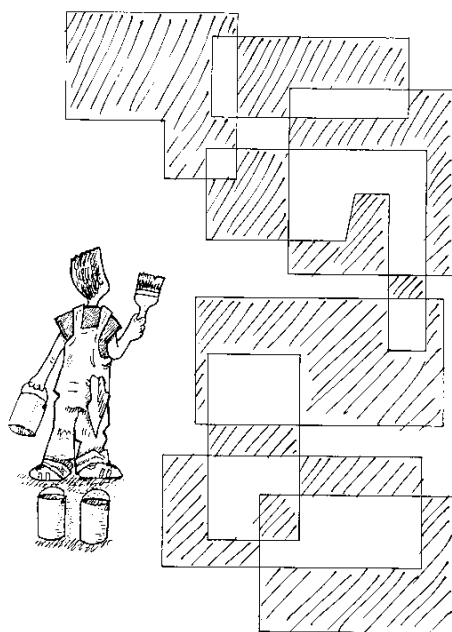


La prima mappa del foglio di lavoro 2 può essere colorata con tre colori mentre per l'altra in basso sono necessari quattro colori. Qui di lato sono proposte due possibili soluzioni.





La mappa del foglio di lavoro 3 è una mappa a tre colori, una soluzione possibile e' mostrata qui a fianco.



Questa è la soluzione per la mappa del foglio di lavoro 4. In questa mappa sono sufficienti due colori (qui rappresentati con il bianco e il tratteggiato).

Attività 15

La città turistica — *Gli insiemi dominanti*

Sommario

Molte situazioni della vita reale possono essere astratte e modellate tramite una rete di “grafi” del medesimo tipo utilizzato per l’attività 14 sulla colorazione. Le reti mostrano moltissime potenzialità per lo sviluppo di algoritmi che siano di uso pratico. Nella presente attività vogliamo marcare qualche giunzione (o “nodo”) in modo che tutti i restanti nodi siano al più ad una distanza di un arco dai nodi marcati. La domanda è: quanti nodi saranno marcati? La questione rappresenta, in realtà, un problema estremamente complesso.

Abilità

- ✓ Costruire mappe
- ✓ Creare relazioni
- ✓ Risolvere rompicapo
- ✓ Ricercare soluzioni in modo iterativo, per passi successivi

Età

- ✓ dai 7 anni in su

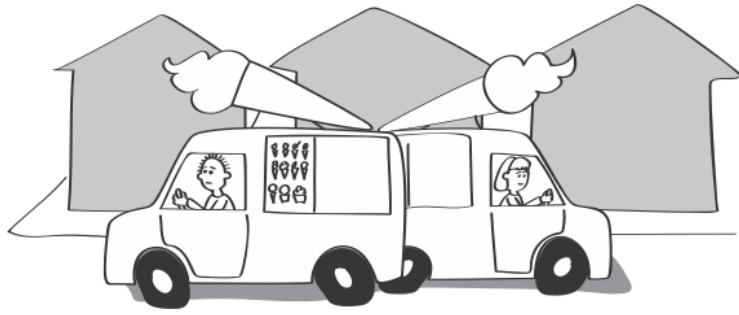
Materiale

Ciascun gruppo di studenti avrà bisogno di:

- ✓ una copia del foglio di lavoro: i furgoncini dei gelati (pag. 183)
- ✓ molti gettoni di due diversi colori.

L’insegnante avrà bisogno di:

- ✓ un’immagine proiettata della soluzione dei furgoncini del gelato, oppure una lavagna dove disegnarla



Insiemi dominanti

Introduzione

Il foglio dove sono disegnati i *furgoncini dei gelati* rappresenta la mappa della città turistica. Le linee sono le strade mentre i nodi rappresentano gli incroci. La città è situata in un Paese con un clima torrido tanto che nella stagione estiva i furgoncini dei gelati sono parcheggiati agli incroci delle strade per vendere gelati ai turisti.

Quello che vorremmo fare è posizionare i furgoncini in modo tale che ogni abitante della città possa raggiungerli camminando fino alla fine della propria strada e, al più, per un altro isolato. Ovviamente, la soluzione sarebbe molto più facile se le persone vivessero negli incroci piuttosto che lungo le strade... Potrebbero comprare un gelato camminando al più un isolato. La domanda, quindi, è: quanti furgoncini sono necessari e su quali incroci devono essere posizionati?

Discussione

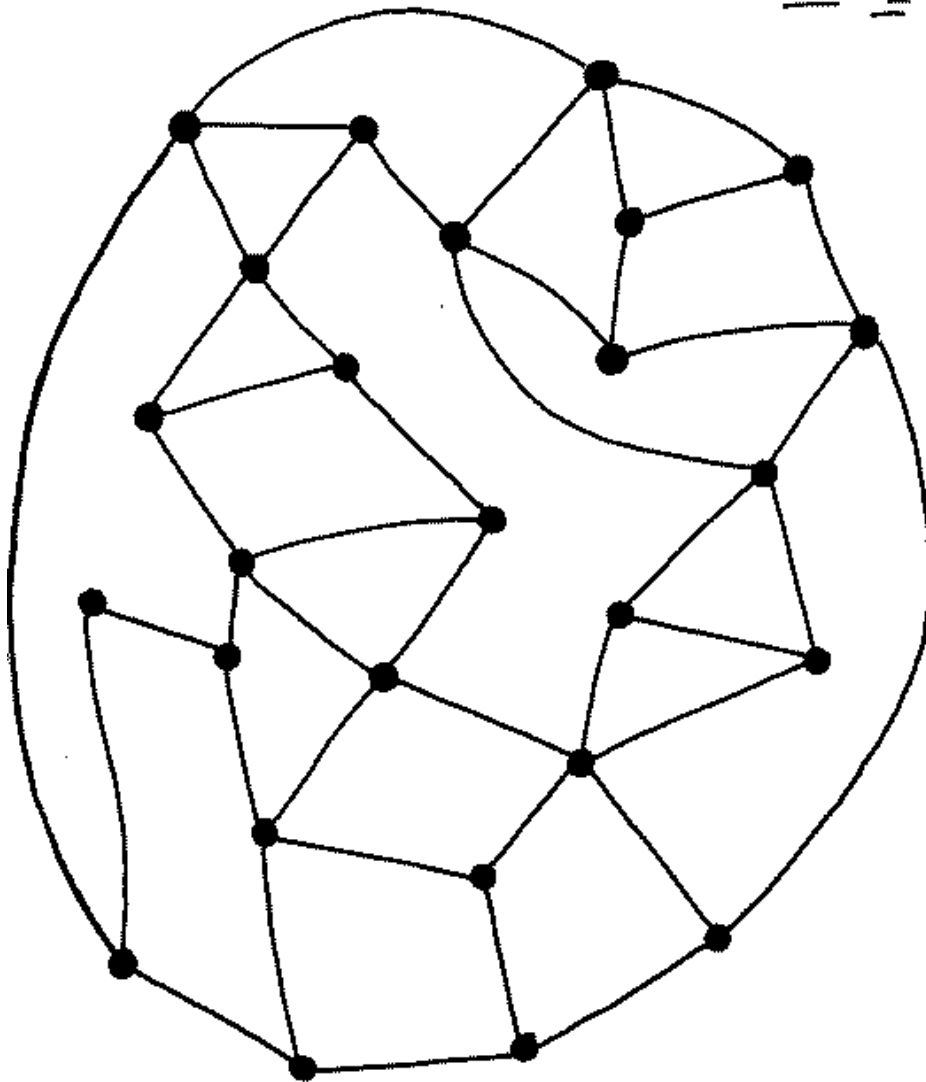
1. Formate piccoli gruppi di studenti e fornite a ciascun gruppo la mappa della città turistica e un po' di gettoni. Spiegate la storia.
2. Mostrate agli studenti come posizionare un gettone su un incrocio identificando, così, un furgoncino dei gelati; poi posizionate altri gettoni diversamente colorati in incroci che siano ad una strada di distanza dai furgoncini. Le persone che abitano in quegli incroci (oppure lungo le strade che vi convergono) sono serviti da quel furgoncino dei gelati.
3. Lasciate sperimentare gli studenti con diverse posizioni dei furgoncini in modo che trovino configurazioni in grado di servire tutte le abitazioni, ricordando loro che i furgoncini sono costosi e l'idea è di averne il meno possibile. È ovvio che le condizioni sono soddisfatte se ci sono abbastanza furgoncini da posizionare in tutti gli incroci. La domanda interessante è: di quanti ne avete bisogno?
4. Il minimo numero di furgoncini per la città turistica è sei. La soluzione è rappresentata qui. Tuttavia è estremamente complesso trovare questa soluzione! Dopo un po' di tempo, spiegate alla classe che sei furgoncini sono sufficienti e sfidate gli studenti a trovare un modo per posizzionarli. Questo è un problema molto complesso: molti gruppi abbandoneranno la sfida. Anche perché è già difficile da trovare una soluzione con otto o nove furgoncini...



5. La mappa intera della città turistica è stata costruita combinando assieme sei mappe ciascuna delle quali ha come soluzione un solo furgoncino dei gelati. Si sono poi congiunte tra loro attraverso tante strade per distogliere l'attenzione dalla soluzione complessiva... L'accorgimento principale è quello di non posizionare ulteriori strade sugli incroci che fanno parte della soluzione (contrassegnati da un cerchio vuoto). Semmai, le strade possono essere congiunte agli altri incroci (il cerchio pieno). Mostrate alla classe questa tecnica.
6. Invitate gli studenti a costruire le loro mappe complesse sfruttando questa tecnica. Potrebbero così sfidare i propri amici o i propri genitori. Scopriranno che possono ideare rompicapo che essi stessi possono risolvere ma non gli altri! Questi problemi sono esempi di ciò che viene identificato col nome di "funzioni a senso unico" *one-way function*: è facile arrivare ad avere un rompicapo che sia estremamente complesso da risolvere, a meno che tu non ne sia l'autore. Le funzioni a senso unico giocano un ruolo cruciale nella crittografia (fai riferimento alle attività 18 e 19).

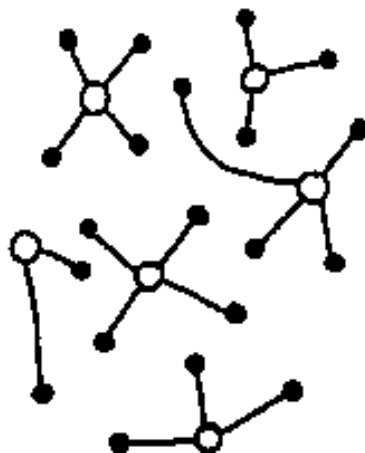
Foglio di lavoro: i furgoncini dei gelati

Comprendi come posizionare i furgoncini dei gelati negli incroci in modo che ogni altro incrocio sia connesso ad uno con un furgoncino.



Foglio di lavoro: soluzione dei furgoncini dei gelati

Mostra questa mappa per far capire com'è stato realizzato il rompicapo.



Variazioni ed estensioni

Ci sono tantissime situazioni della vita reale che generano questa tipologia di problemi: posizionare buche delle lettere, idranti per i vigili del fuoco, stazioni della polizia e così via. Ma nella vita reale la mappa non sarà costruita utilizzando trucchi che la rendano semplice. Se realmente c'è la necessità di risolvere un problema come questo, come si fa?

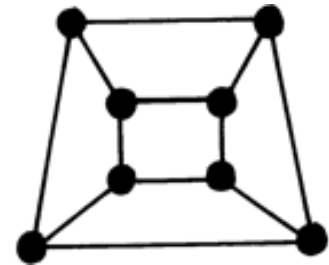
C'è una via molto molto facile: si prendono in considerazione tutti i possibili modi di posizionare i furgoncini dei gelati e si controlla qual è il risultato migliore. Con i 26 incroci che troviamo nella città turistica ci sono 26 modi di piazzare un furgoncino. È facile verificare tutte le 26 possibilità ed è ovvio che nessuna soddisferà la condizione desiderata. Con due furgoncini, invece, ci sono 26 luoghi dove posizionare il primo ed altri 25 dove posizionare il secondo (ovviamente non si posizioneranno entrambi nel medesimo incrocio): $26 \times 25 = 650$ possibilità da controllare. In ogni caso, l'operazione di controllo è semplice, sebbene sia abbastanza seccante controllare tutte le combinazioni. In realtà, si avrà bisogno di controllare solo metà delle combinazioni totali (325), in quanto se il furgoncino 1 è all'incrocio A e il furgoncino 2 nel B non c'è necessità di controllare se il furgoncino 2 è all'incrocio A e il furgoncino 1 nel B. Si può proseguire con tre furgoncini (2600 possibilità) o quattro (14950) e così via. Chiaramente, 26 furgoncini sono abbastanza perché ci sono solo 26 incroci e non si possono avere più furgoncini in un incrocio. Un altro modo di arrivare a quel numero è quello di considerare il numero totale di configurazioni con 26 incroci e un numero qualsiasi di furgoncini. Dal momento che ci sono due possibilità per ogni incrocio di avere o non avere un furgoncino, significa che le configurazioni sono 2 elevato alla 26 cioè 67.108.864.

Questo metodo di risoluzione è chiamato algoritmo *brute-force* e richiede un tempo molto lungo per la soluzione. È ampiamente accettata la credenza che i computer siano così veloci che possano risolvere qualsiasi problema rapidamente. L'efficienza dell'algoritmo *brute-force* dipende da quanto velocemente riesce a verificare se una particolare configurazione è una soluzione. Per far ciò occorre controllare ogni incrocio per trovare la distanza del furgoncino più vicino. Si supponga che un'intera configurazione possa essere verificata in un secondo. Quanto si impiegherà a verificare tutte le 2 elevato 26 possibilità? (Risposta: quelle combinazioni sono circa 67 milioni; ci sono 86.400 secondi in un giorno e 2 elevato alla 26 secondi sono circa 777 giorni, cioè più o meno due anni). Supponendo che invece di un secondo ci si impieghi un millesimo di secondo, allora in due anni si potranno verificare mappe con 36 incroci (perché 2 elevato alla 36 è circa 1000 volte 2 elevato alla 26). Ma anche se il computer fosse un milione di volte più veloce, così che si possano verificare un milione di

configurazioni ogni secondo, allora ci si impiegherebbero due anni per lavorare su una mappa con 46 incroci. E queste non sono città particolarmente grandi! (A proposito, quanti incroci ha la tua città?)

Dal momento che l'algoritmo è lento, Ci sono altri modi per risolvere il problema? Beh, potremmo provare l'approccio "greedy" che riusciva a risolvere il problema nella "città fangosa" (attività 9). Abbiamo bisogno di pensare a come essere greedy con i gelati. Cioè come applicare l'approccio al problema dei furgoncini dei gelati. Il modo per farlo è quello di inserire il primo furgone all'incrocio che collega il maggior numero di strade, il secondo furgoncino al successivo incrocio più connesso e così via. Tuttavia, questo non produce necessariamente un insieme minimo di furgoncini in realtà, l'incrocio con più strade nella città turistica, che ne ha cinque, non è un buon posto per piazzare un furgone (controlla con la classe).

Diamo un'occhiata a un problema più facile. Invece di chiedere di trovare una configurazione minima, si supponga di avere già una configurazione e ci si chieda se è minima o no. In alcuni casi, la risposta è facile. Per esempio, questo diagramma mostra una mappa molto più semplice la cui soluzione è abbastanza immediata. Se immaginate le strade come bordi di un cubo, è chiaro che due furgoni gelato nei vertici diagonalmente opposti del cubo sono sufficienti. Inoltre, dovrete essere in grado di convincervi che non è possibile risolvere il problema con meno di due furgoncini. È molto più difficile, se non impossibile, convincersi che la città turistica non può essere servita da meno di sei furgoni. Per mappe generiche è estremamente difficile dimostrare che una certa configurazione di furgoncini gelato sia minima.



Cosa c'entra tutto questo?

Una delle cose interessanti del problema dei gelati è che nessuno *sa* se esista un algoritmo per la ricerca di un insieme minimo di posizioni che sia significativamente più veloce rispetto al metodo *brute force* (provare una ad una tutte le soluzioni possibili)! Il tempo impiegato dal metodo *brute force* cresce esponenzialmente con il numero di intersezioni: si chiama algoritmo tempo-esponenziale. Un algoritmo tempo-polinomiale è invece un metodo in cui il tempo di esecuzione cresce come il quadrato o il cubo o la diciassettesima potenza o qualsiasi altra potenza del numero di incroci. Un algoritmo tempo-polinomiale sarà sempre più veloce di un algoritmo esponenziale per mappe sufficientemente grandi. Anche, ad esempio, un algoritmo di grado 17 sarà più veloce, dato che un algoritmo esponenziale supera qualsiasi algoritmo polinomiale una volta che la variabile di riferimento (l'argomento) diventi sufficientemente grande (ad esempio, riscontra come ogni volta che n è più grande di 117 allora n elevato alla 17 è minore di 2 elevato alla n). Esiste un algoritmo polinomiale per trovare il set minimo di posizioni? Nessuno lo sa: anche se molti ricercatori lo hanno cercato. Lo stesso vale per l'apparentemente facile compito di verificare se un particolare insieme di posizioni sia minimo: l'algoritmo *brute-force* di provare tutte le possibilità per i set più ridotti di posizioni è esponenziale nel numero di incroci; invece non sono né stati scoperti né si è dimostrato che esistano algoritmi polinomiali.

Questo vi ricorda l'attività della colorazione (Attività 14)? Dovrebbe. La questione dei furgoncini gelato, che è denominata problema del "minimo insieme dominante", è solo uno di un gran numero, migliaia, di problemi per i quali non è noto se esistono algoritmi polinomiali; questi problemi sono di logica, di ricostruzione di puzzle, di colorazione di mappe, di ricerca di percorsi ottimali sulla mappa, di processi di pianificazione. Sorprendentemente, tutti questi problemi hanno dimostrato di essere equivalenti, nel senso che se un algoritmo polinomiale viene trovato per uno di essi, può essere convertito in un algoritmo polinomiale per tutti gli altri. Si potrebbe dire che hanno un... destino comune.

Questi problemi sono chiamati *NP-completi*. *NP* sta per "non-deterministico polinomiale". Questa denominazione significa che il problema potrebbe essere risolto in un ragionevole lasso di tempo se si avesse a disposizione un computer in grado di provare un arbitrariamente grande numero di soluzioni in una sola volta (che è la parte non-deterministica). Si potrebbe pensare che questo sia un presupposto piuttosto irrealistico e in effetti lo è. Non è possibile costruire questo tipo di computer, in quanto dovrebbe essere arbitrariamente grande! Tuttavia, il concetto di tale macchina è importante come principio, perché sembra che i problemi *NP-completi*

non possano essere risolti in un ragionevole lasso di tempo senza un computer non deterministico.

Inoltre, questo gruppo di problemi si chiama *completo* perché anche se i problemi appaiano molto diversi (per esempio, il problem della colorazione della mappa è molto diverso dal problema di collocare furgoncini gelato) in realtà si può dimostrare che se esiste un metodo efficace di risolvere uno di essi, allora questo metodo può essere adattato per risolvere tutti gli altri problemi *NP-completi*. Questo è quello che intendiamo per "destino comune."

Ci sono migliaia di problemi *NP-completi* e i ricercatori da diversi decenni li stanno studiando, alla ricerca di soluzioni efficienti, senza successo. Se una soluzione efficiente fosse stata scoperta per uno solo di essi, allora avremmo soluzioni efficienti per tutti. Per questo motivo, c'è il forte sospetto che non esista una soluzione efficace. La dimostrazione sugli algoritmi a tempo esponenziale è oggi la questione aperta più rilevante nell'informatica teorica e in tutta la matematica.

Ulteriori approfondimenti

Il libro di Harel, *Algorithmics* [5], introduce parecchi problemi *NP-completi* e discute la questione se esistano algoritmi polinomiali nel tempo. Il libro di Dewdney, *Turing Omnibus* [3], discute la *NP-completezza*. Il testo classico sull'argomento è di Garey & Johnson, *Computers e Intrattabilità* [4], che introduce diverse centinaia di problemi *NP-completi* assieme a tecniche per dimostrare la *NP-completezza*. Tuttavia, è abbastanza complesso ed è adatto solo per specialisti.

Attività 16

Strade ghiacciate — *Gli alberi di Steiner*

Sommario

Talvolta una piccola, apparentemente insignificante, variazione della specifica di un problema può comportare una differenza enorme e rendere lo stesso problema più difficile da risolvere. La seguente attività, in modo simile a quanto già visto nel problema della Città Fangosa (Attività 9), consiste nel trovare percorsi brevi attraverso alcune reti. La differenza è che qui ci è consentito introdurre nuovi punti nella rete, se questi riducono la lunghezza del percorso. Il risultato è un problema ben più difficile che non è legato alla Città Fangosa, ma è algoritmicamente equivalente a quello del Cartografo (Attività 14) e a quello della Città Turistica (Attività 15).

Abilità

- ✓ Visualizzazione spaziale
- ✓ Ragionamento geometrico
- ✓ Procedure algoritmiche e complessità

Età

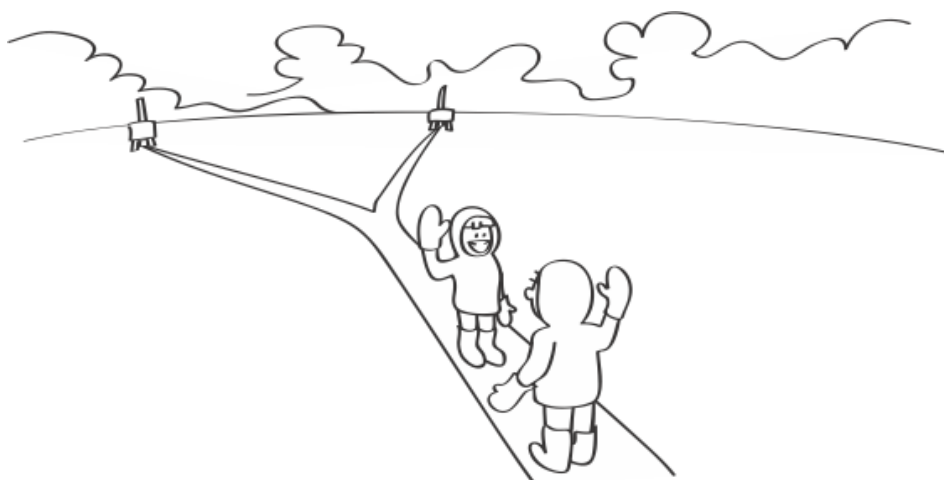
- ✓ dai 7 anni in su

Materiali

Ciascun gruppo di studenti avrà bisogno di:

- ✓ cinque o sei pioli da posizionare a terra (i picchetti vanno bene e le grucce altrettanto),
- ✓ molti metri di elastico o stringa,
- ✓ un righello o un metro a nastro,
- ✓ carta e penna per prendere nota.

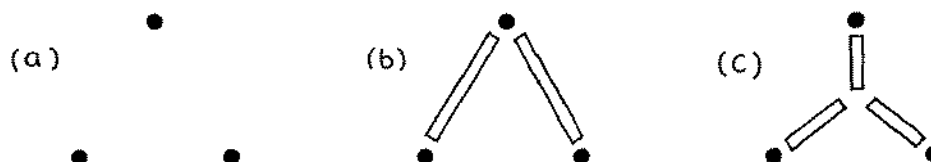
Strade ghiacciate



Introduzione

L'attività precedente, la Città Turistica, ha avuto luogo in un paese molto caldo; questa attività è esattamente l'opposto. Nel gelido nord del Canada (così va la storia), in inverno, sugli enormi laghi ghiacciati, gli spazzaneve costruiscono strade per collegare i siti di perforazione in modo che gli equipaggi possano farsi visita a vicenda. Con quel freddo vogliono costruire il minor numero di strade; il vostro compito è quello di capire dove costruire effettivamente le strade. Non ci sono vincoli: le strade possono passare in qualsiasi punto dei laghi congelati. È tutto piatto.

Le strade dovrebbero ovviamente essere rettilinee, poiché l'introduzione di curve non farebbe che aumentare la lunghezza inutilmente. Ma non è così semplice collegare tutti i siti con linee rette: l'aggiunta di intersezioni lungo le lande ghiacciate a volte potrebbe ridurre la lunghezza stradale totale ed è la lunghezza totale, alla fine, ad essere importante, non il tempo di viaggio da un sito all'altro.



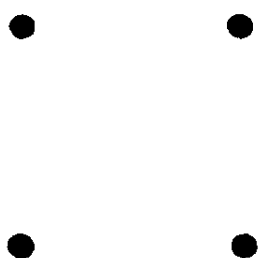
In questa figura, (a) mostra tre siti di perforazione. Collegare uno di loro a ciascuno degli altri (come in (b)) formerebbe una rete stradale accettabile. Altra possibilità è creare un'intersezione da qualche parte vicino al centro del triangolo e collegarlo ai tre siti (c). Se si misura la lunghezza totale di strada della strada ne deriva che questa è davvero una soluzione migliore. L'intersezione aggiunta si chiama un punto di "Steiner" dal matematico svizzero Jacob Steiner (1796-1863), che ha

formalizzato il problema e che stato il primo a notare che la lunghezza totale può essere ridotta introducendo nuovi punti. Potreste pensare ad un punto di Steiner come ad nuovo, fittizio, sito di perforazione.

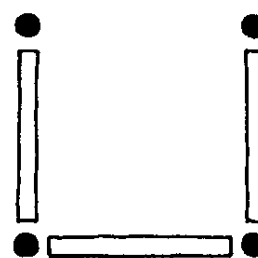
Discussione

Descrivete il problema su cui gli studenti lavoreranno. Attraverso l'esempio precedente, dimostrate agli studenti partendo da tre siti, aggiungendone uno, talvolta la soluzione migliora riducendo la quantità di strade da costruire.

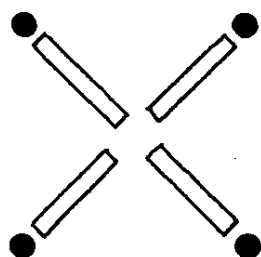
(a)



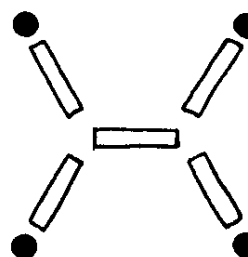
(b)



(c)



(d)

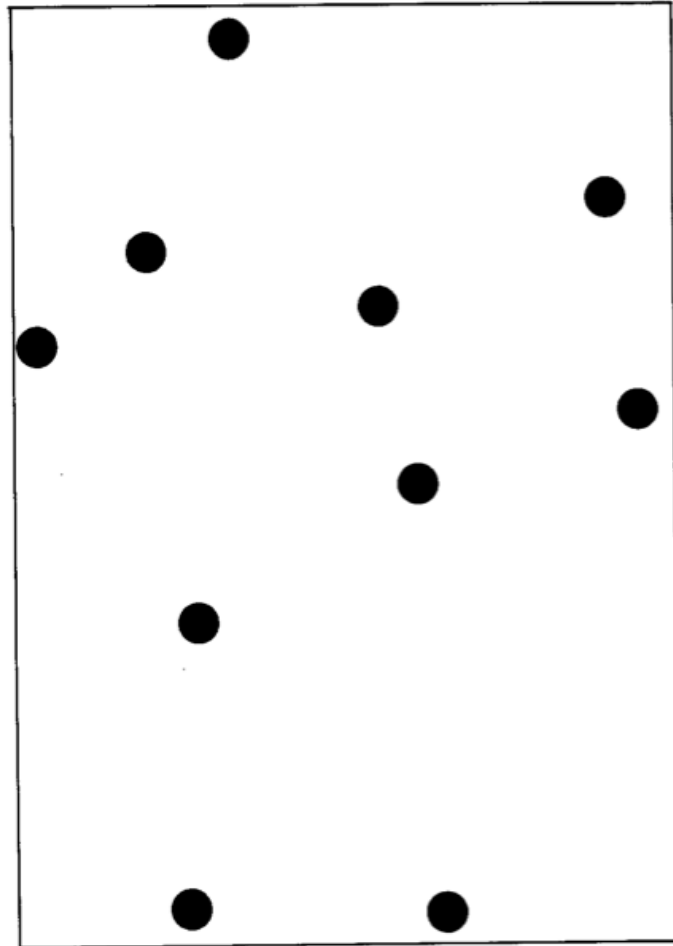


1. Gli studenti useranno quattro punti disposti in una piazza, come illustrato in (a). Andate in un giardino e ciascun gruppo collochi quattro pioli nell'erba formando un quadrato di circa 1 metro per 1 metro.
2. Fate in modo che gli studenti sperimentino, collegando i pioli con stringa o con l'elastico, misurando e registrando la lunghezza totale minima. In questa fase non dovrebbero usare i punti di Steiner. (Il minimo si ottiene collegando tre lati del quadrato, come in (b) e la lunghezza totale è di 3 metri).
3. Ora vediamo se gli studenti riescono a fare di meglio utilizzando un punto di Steiner. (Il posto migliore è al centro della piazza come illustrato in (c)). Quindi la lunghezza totale è di $2\sqrt{2} = 2.83$ metri. Suggeste che si potrebbe fare ancora di meglio con due punti di Steiner. (In effetti posizionando i due punti come in (d), formando

angoli di 120 gradi tra le strade in entrata, la lunghezza totale è di $1 + \sqrt{3} = 2,73$ metri).

4. Possono gli studenti fare di meglio con tre punti di Steiner? (No - due punti sono il meglio e nessun vantaggio si acquisisce utilizzandone di più).
5. Discutete con gli studenti perché questi problemi sembrano difficili (È perché non si sa dove mettere i punti di Steiner e ci sono un sacco di possibilità da provare).

Attività: Albero di Steiner - Esempio 1



Page 2

Attività: Albero di Steiner - Esempio 2

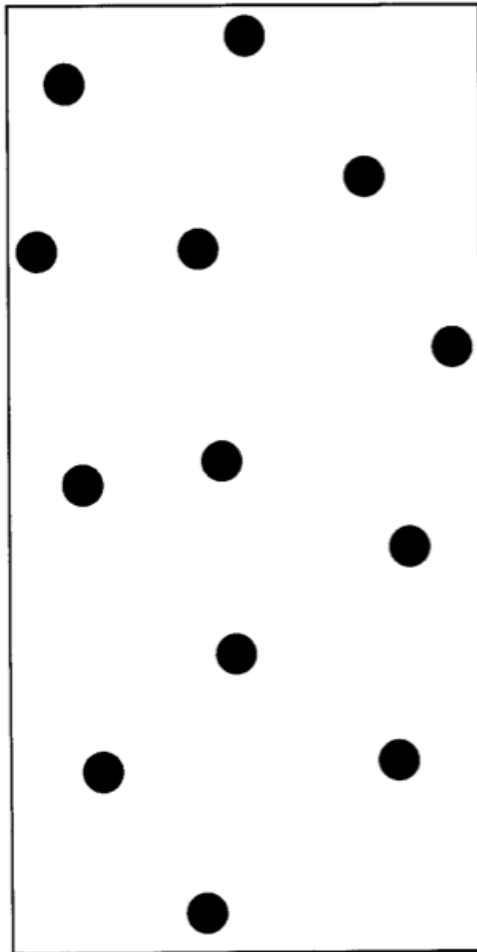
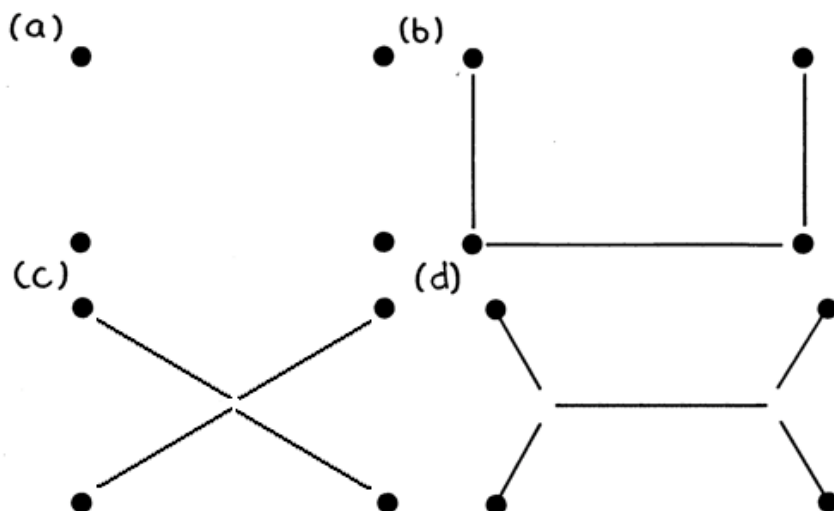


Diagram 132

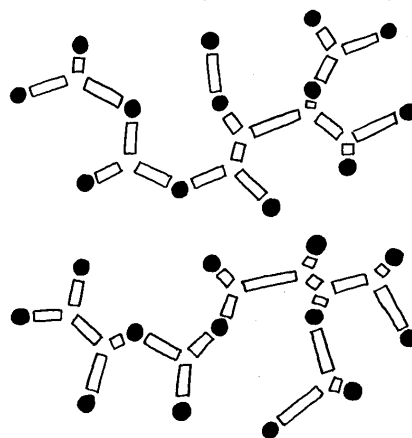
Variazioni ed estensioni



1. Un esperimento interessante per i gruppi che terminano prima l'attività precedente consiste nel lavorare con un rettangolo di circa 1 metro di 2 metri (a). Gli studenti troveranno che l'aggiunta di un punto di Steiner peggiora le cose, ma due producono un miglioramento. (Le lunghezze sono 4 metri per (b), $2\sqrt{5} = 4.47$ metri per (c) e $2 + \sqrt{3} = 3.73$ metri per (d)). Vedi se possono capire perché la configurazione con un punto peggiora le cose per i rettangoli e non per le piazze. (È perché quando il quadrato viene allungato in un rettangolo, la quantità di "stiramento" viene aggiunta solo una volta in (b) e (d), ma entrambe le diagonali aumentano in (c)).

2. Gli studenti più anziani possono lavorare su un problema più complicato. Sono due i layout di siti da collegare con strade di ghiaccio riportati nei fogli di lavoro. Gli studenti possono sperimentare diverse soluzioni sia utilizzando nuove copie del foglio di lavoro sia scrivendo su un foglio trasparente posto sopra i fogli di lavoro di Steiner. In alternativa, le mappe possono essere tracciate nel terreno utilizzando i pioli. L'esperienza finisce quando gli studenti pensano di aver stabilito un nuovo record per la distanza più breve. (Le figure a destra mostrano la soluzione minima per il primo esempio e due possibili soluzioni per il secondo, il cui totale della lunghezza è abbastanza simile). Il fatto che ci siano due soluzioni simili ben giustifica perché questi tipi di problemi siano così difficili: ci sono tante scelte dove poter mettere i punti di Steiner!

Due alberi di Steiner per il secondo esempio

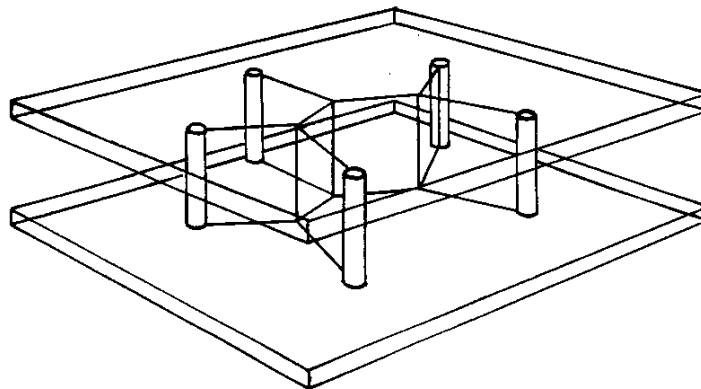




3. Reti a “scala” come questa forniscono un altro modo per estendere il problema.
4. Alcuni alberi di Steiner minimi per le reti a scala sono riportati qui di seguito.

La soluzione per una scala a due pioli è la stessa del quadrato. Tuttavia, per una scala a tre pioli la soluzione è molto diversa, come scoprirete ricorrendo alla memoria! La soluzione per quattro pioli si ottiene congiungendo assieme quelle per le scale a due pioli, mentre la soluzione per cinque pioli assomiglia più al prolungamento della soluzione per il caso a tre pioli. In generale, la forma dell’albero di Steiner minimo per una scala dipende dal fatto che la scala abbia un numero pari o dispari di pioli. Se è pari allora è come se diverse scalette a due pioli fossero state congiunte. In caso contrario, è come una ripetizione della scala a tre gradini. Ma provare queste cose con rigore non è facile.

Un’altra interessante attività è quella di costruire modelli a bolla-di-sapone per gli alberi di Steiner. È possibile farlo prendendo due fogli di plastica rigida trasparente e inserendo dei perni tra loro per rappresentare i siti da attraversare, come mostrato qui.



Ora immergete il tutto in una soluzione di sapone. Quando emerge scoprirete che un film di sapone collega i perni con uno stupendo albero-rete di Steiner.

Purtroppo, però, non è necessariamente un albero di Steiner *minimo*. La pellicola di sapone identifica una configurazione che minimizza la lunghezza totale, ma il minimo è soltanto locale, non

necessariamente globale. Potrebbero esserci modi completamente diversi per collocare i punti di Steiner e per ottenere una lunghezza totale inferiore. Ad esempio, si può immaginare che la pellicola di sapone acquisisca quando emerge dal liquido la disposizione sia della prima che della seconda configurazione dell'estensione 2, in due casi diversi!

Cosa c'entra tutto questo?

Le reti su cui abbiamo lavorato sono degli alberi di Steiner minimi. Sono denominate "alberi" perché non hanno cicli, proprio come i rami di un albero vero, che crescono singolarmente ma non si ricongiungono (normalmente) per crescere assieme. Sono dette alberi di "Steiner" perché nuovi punti, i punti di Steiner, possono essere aggiunti ai luoghi connessi dagli alberi. E si chiamano "minimi" perché fra tutti gli alberi che collegano tali luoghi hanno la lunghezza minima. Nella Città Fangosa (Attività 9) abbiamo appreso che una rete che collega una serie di siti minimizzando la lunghezza totale è chiamata *spanning tree* minimo: gli alberi di Steiner sono la stessa cosa tranne che per il fatto che nuovi punti possono essere introdotti.

È interessante sapere che, mentre c'è un algoritmo molto efficiente per trovare lo *spanning tree* minimo (Attività 9), ovvero un algoritmo greedy che agisce ripetutamente connettendo i due punti più vicini non ancora connessi, non c'è in generale una soluzione efficiente al problema di Steiner. Gli alberi di Steiner sono molto più difficili da costruire perché si deve decidere dove mettere i punti extra. Piuttosto sorprendentemente, la parte difficile del problema dell'albero di Steiner non consiste nel determinare la posizione precisa dei punti di Steiner, ma nel decidere approssimativamente dove metterli: per esempio questa è la differenza tra le due soluzioni all'Esempio 2. Una volta che sapete in quali regioni posizionare i nuovi punti, trovare la loro posizione ottimale è relativamente semplice. Le pellicole di sapone lo fanno in modo molto efficace e allo stesso modo riescono i computer.

Trovare alberi di Steiner minimi è parte di una storia che permette grandi risparmi agli operatori telefonici. Prima del 1967, le aziende negli Stati Uniti utilizzavano grandi reti telefoniche private, affittavano le linee da una compagnia telefonica per interconnettere le loro sedi. L'importo fatturato non veniva calcolato sulla base di come effettivamente erano utilizzati i collegamenti, ma sulla base della rete più breve necessaria. Il ragionamento era che il cliente non avrebbe dovuto pagare un extra solo perché la società telefonica utilizza un percorso inefficiente. In origine, l'algoritmo calcolava quanto far pagare determinando l'albero di copertura minimo (*spanning tree*). Tuttavia, intorno al 1967 fu notato, da una compagnia aerea che aveva tre grandi sedi, che se la società avesse avuto una quarta sede in un punto intermedio allora la lunghezza totale della rete si sarebbe ridotta. La compagnia telefonica fu costretta a ridurre gli addebiti a quelli relativi al percorso minimo, come se ci fosse stata una centrale telefonica nel punto di Steiner! Anche se, per le configurazioni tipiche, l'albero di Steiner minimo è solo il 5% o il 10% più corto del minimo *spanning tree*, questo può essere un notevole risparmio nel caso di grandi quantità di denaro. Il problema dell'albero di Steiner è talvolta chiamato il "problema della rete più

breve” perché consiste nel trovare la rete più breve che collega un insieme di siti.

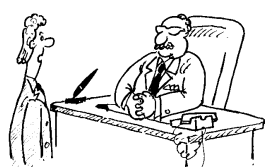
Se avete affrontato le due attività precedenti, il Cartografo Povero e la Città Turistica (rispettivamente attività 14 e 15), non sarete sorpresi di sentire che il problema dell'albero minimo di Steiner è *NP-completo*. Se il numero di siti aumenta, anche il numero di possibili posizioni per punti di Steiner aumenta e la ricerca in tutte le possibilità comporta una ricerca esponenziale. Questo è un altro fra le migliaia di problemi per cui semplicemente non è noto se la ricerca esponenziale sia il meglio che si possa fare o se vi sia ancora da scoprire un algoritmo polinomiale. Ciò che si sa, però, è che se un algoritmo polinomiale viene trovato per questo problema esso può essere trasformato in un algoritmo polinomiale anche per la colorazione del grafo, per la ricerca del minimo insieme dominante e per tutti gli altri problemi nell'insieme *NP-completi*.

Abbiamo spiegato alla fine della precedente attività che “NP” in *NP-completo* sta per “polinomiale non deterministico,” e *completo* si riferisce al fatto che se un algoritmo polinomiale viene trovato per uno dei problemi *NP-completi*, allora quell'algoritmo può essere trasformato in un algoritmo polinomiale per ciascuno degli altri. L'insieme dei problemi che sono risolvibili in tempo polinomiale è chiamato P. Così la domanda cruciale è: esistono algoritmi polinomiali per problemi *NP-completi*, in altre parole, è $P = NP$? La risposta a questa domanda non è nota e rappresenta uno dei grandi misteri dell'informatica moderna.

I problemi per i quali questi algoritmi esistono, anche se questi algoritmi potrebbero essere piuttosto lenti, sono chiamati “trattabili”. I problemi per i quali non esistono sono chiamati “intrattabili”, perché non importa quanto veloce sia il vostro computer o quanti computer utilizzate insieme: un piccolo aumento nelle dimensioni del problema comporta che questo non possa più essere risolto in pratica. Non è noto se i problemi NP-completi, che includono la città turistica, il cartografo povero e le strade ghiacciate siano trattabili o no. Ma la maggior parte degli studiosi informatici sono pessimisti sul fatto che un algoritmo polinomiale si potrà mai trovare per problemi *NP-completi*, così dimostrando che un problema *NP-completo* rappresenta una prova evidente che il problema è di per sé intrattabile.

Cosa si può fare quando il vostro capo vi chiede di elaborare un algoritmo efficiente per ricercare la soluzione ottimale per un problema e non riuscite a trovarne uno? (Come certamente è accaduto quando la compagnia aerea ha dimostrato il fatto che i costi di rete potevano essere ridotti con l'introduzione di punti di Steiner). Sarebbe bello se si potesse dimostrare che non esiste un algoritmo efficiente per trovare la soluzione ottimale. Ma è molto difficile dimostrare risultati negativi di questo tipo in informatica: magari qualche programmatore particolarmente abile potrebbe in futuro sviluppare un trucco originale

che risolva il problema. Così, purtroppo, è improbabile essere in grado di dire con certezza e categoricamente che non esista nessun algoritmo efficiente e che il problema sia quindi intrattabile. Ma se è possibile dimostrare che il problema è *NP-completo*, allora è vero che migliaia di persone nei laboratori di ricerca hanno lavorato su problemi che in realtà sono equivalenti al vostro e anche loro non sono riusciti a trovare una soluzione efficiente. Mal comune...



"Non riesco a trovare un algoritmo efficiente, credo di essere solo troppo stupido."



"Non riesco a trovare un efficiente algoritmo, perché tale algoritmo non è possibile."



"Non riesco a trovare un algoritmo efficiente, ma nemmeno hanno potuto tutti questi personaggi famosi."

Cosa fare quando non riesci a trovare un algoritmo efficiente: tre possibilità

Naturalmente, nella vita reale questi problemi devono ancora essere risolti e in quel caso le persone si rivolgono a soluzioni euristiche: algoritmi che non garantiscono la miglior soluzione possibile, ma forniscono una soluzione che si discosta di poco da quella ottimale, consentendo inoltre di valutare e limitare l'errore. Gli algoritmi euristici possono risultare molto veloci e la differenza con la soluzione ottimale non essenziale all'atto pratico, quindi possono risultare accettabili per proseguire con il lavoro. È solo frustrante sapere che potrebbero esistere migliori calendari o migliori percorsi di reti o strade.

Ulteriori letture

Il fumetto riprodotto qui sopra è basato su di un libro classico di Garey e Johnson *Computer e Intrattabilità* [4].

La sezione "ri-creazioni al Computer" di *Le Scienze* (traduzione di *Scientific American*), nel fascicolo di giugno 1984 contiene una breve descrizione di come costruire gli alberi di Steiner con bolle di sapone, assieme a interessanti descrizioni di altri gadget analogici per la soluzione dei problemi, tra cui un computer di spaghetti per l'ordinamento, la culla di un gatto di stringhe per la ricerca dei percorsi più brevi in un grafico e un dispositivo a specchi e luci per verificare se un numero è primo o meno. Questi esempi appaiono anche in una specifica sezione riguardante i computer analogici nel lavoro di Dewdney *Turing Omnibus* [3].

Parte V

Condividere segreti e combattere il crimine — *la Crittografia*

Condividere segreti e combattere il crimine

Sicuramente avrete sentito parlare di spie e agenti segreti che usano codici cifrati e magiche scritture invisibili per scambiare messaggi. È per questo che è nata la "crittografia", l'arte di scrivere e di decifrare codici segreti. Durante la seconda guerra mondiale, gli inglesi crearono una macchina specificatamente disegnata per decifrare i codici militari. Poi venne l'era dei computer e cambiò ogni cosa e la crittografia entrò in una nuova era. Una quantità enorme di calcoli, inimmaginabile precedentemente, poté essere utilizzata per decifrare i codici. Quando le persone iniziarono a condividere l'uso di computer fra loro, le parole d'ordine o *password* iniziarono ad essere utilizzate non per entrare in una base militare ma per proteggere la riservatezza delle informazioni. Quando poi i computer vennero interconnessi da reti, ci furono ulteriori motivi per proteggere le informazioni da quanti avrebbero voluto leggerle in modo non autorizzato. Con la posta elettronica poi si pose il problema di certificare che chi firma un messaggio sia veramente chi dice di essere. Ora le persone fanno transazioni bancarie *on-line*, tramite servizi Web, sulla rete Internet. Oggi si acquistano e si vendono merci usando i computer. E' necessario per questo che gli strumenti usati per spedire gli ordini e per trasferire il denaro siano sicuri. Il crescente pericolo di un attacco terroristico tramite l'uso di computer rende il problema della sicurezza informatica sempre più importante.

La crittografia probabilmente vi farà pensare a *password* segrete e a come miscugliare le lettere di un messaggio in modo che il nemico non possa leggerlo. La realtà è molto differente. I sistemi moderni di elaborazione non memorizzano le *password* segrete, perché se lo facessero chiunque riuscisse ad averne accesso potrebbe violare completamente la sicurezza del sistema. Questo sarebbe disastroso: potrebbero fare false operazioni bancarie, spedire messaggi fingendo di essere qualcun altro, leggere file segreti, dare ordini ad eserciti, far cadere governi. Oggi le password sono gestite tramite le "funzioni a senso unico" (*one-way function*) di cui abbiamo parlato nell'attività 15. E per criptare i messaggi non si mischiano semplicemente le lettere del messaggio: si usano tecniche basate su problemi molto difficili da risolvere, i problemi "intrattabili" che abbiamo introdotto nella parte IV.

In questa sezione scopriremo un modo semplice per calcolare l'età media delle persone di un gruppo senza che nessuno del gruppo debba dichiarare la propria. Imparerete come due persone che non si fidano l'una dell'altra, possano giocare a testa o croce ed essere certe della correttezza del risultato sebbene abitino in città diverse e quindi non possano usare la classica monetina. Scoprirete poi un metodo per codificare messaggi segreti in modo che una sola persona possa

decodificarli anche se il metodo di codifica è noto a tutti.

Per gli insegnanti

Le attività che seguono forniscono allo studente una esperienza pratica sulle moderne tecniche di crittografia digitale, che sono molto differenti da quanto si sente dire comunemente in giro. Ci sono innanzitutto due concetti chiave. Il primo è la nozione di “protocollo”, che è una definizione formale di una transazione. Il protocollo può farci ricordare il lavoro dei diplomatici, che devono rispettare una certa *etiquette* nelle loro parole ed azioni. Anche i computer hanno le loro regole di *etiquette* da rispettare! Compiti che appaiono difficilissimi possono essere portati a compimento da protocolli sorprendentemente semplici. L'attività 17, che può essere svolta in pochi minuti, mostra come un insieme di persone possano, di comune accordo, calcolare l'età (o il reddito) medio del gruppo senza che alcuno di loro debba dichiarare ad altri la propria età (o il proprio reddito). Il secondo concetto è che la complessità computazionale, l'intrattabilità, può ricoprire un ruolo fondamentale per proteggere il dialogo con altri attraverso l'uso di computer. L'attività 18 mostra come due persone, che non necessariamente si fidino l'una dell'altra, possano concordare sul risultato di una partita a testa o croce sebbene possano parlare solo al telefono (questa attività introduce anche i concetti di algebra di Boole e di circuiti logici, che possono essere approfonditi in altri percorsi didattici). L'attività 19, infine, mostra come le tecniche crittografiche consentano di criptare messaggi in modo sicuro anche se il metodo usato è conosciuto pubblicamente.

Alcune di queste attività, in modo particolare l'ultima, richiedono uno sforzo non banale. È necessario che motivate la vostra classe giocando sul senso di meraviglia che accompagna la risoluzione di problemi che sembrano impossibili. È necessario creare questo stupore e fermarsi ripetutamente durante l'esposizione dell'attività per fare in modo che gli studenti non perdano la visione della (affascinante) luna perché osservano solo il (faticoso) dito. Queste attività sono fra le sfide più ardite di questo libro e sono anche le più tecnicamente intricate. Se ritenete che siano troppo complesse, potete saltare direttamente alla successiva parte VI, che ha una natura completamente differente, non tecnica.

Per le menti più tecno-curiose

Man mano che i computer invadono le nostre vite quotidiane, l'uso e le reali finalità della crittografia possono diventare non chiare. Molte persone non sono consapevoli delle capacità dei moderni metodi di crittografia. Il risultato è che quando grandi istituzioni, siano esse pubbliche o private, configurano sistemi che elaborano informazioni personali, tendono a delegare a tecnocrati le scelte chiave su come le informazioni verranno richieste, su quali verranno elaborate, su quali verranno fornite e a chi. Se le persone comprendessero meglio le

possibilità consentite dalla tecnologia moderna, potrebbero partecipare più attivamente a queste decisioni e la società potrebbe avere una differente, più equa infrastruttura di gestione delle informazioni

Gli argomenti trattati in questa sezione del testo ovvero i protocolli di gestione dell'informazione nascosta, i protocolli crittografici, la crittografia a chiave pubblica, sono generalmente considerati contenuti avanzati dei corsi di informatica. Ma le idee alla base non sono difficili da comprendere. Sono gli aspetti tecnici, non i concetti sottostanti, ad essere complessi. Nelle situazioni pratiche relative all'e-commerce, per esempio, gli aspetti tecnici sono nascosti all'interno del software che rende molto semplice l'uso di questi strumenti. Ma è molto importante comprendere le idee che consentono a questi strumenti di funzionare, per poter aver la consapevolezza di cosa possa essere fatto e cosa no.

I sistemi crittografici sono di grande interesse per i governi, non solo perché vogliono mantenere la sicurezza delle loro comunicazioni ufficiali, ma anche per la preoccupazione che la comunicazione criptata possa essere usata da persone coinvolte in attività illegali come il traffico di droga o il terrorismo. Le intercettazioni delle comunicazioni di queste persone possono diventare inutili se non c'è modo di decodificare i loro messaggi. Queste preoccupazioni hanno creato un vasto dibattito fra chi deve mantenere la legalità (che vuole limitare la inviolabilità dei sistemi crittografici) e chi difende le libertà civili (che non accetta che il governo possa aver accesso alle comunicazioni private). Il governo degli Stati Uniti per un certo periodo impose restrizioni all'uso di alcuni metodi crittografici assimilandoli ad armi, come se fossero bombe o fucili. Chiunque può creare un canale di comunicazione sicuro, a patto che abbia le informazioni corrette e la capacità tecnica, ma questi strumenti possono essere pericolosi se posti nelle mani sbagliate. Ad un certo punto ci fu anche un esteso dibattito in merito al "Clipper Chip", un circuito integrato per la crittografia che avrebbe dovuto gestire una password aggiuntiva chiamata *chiave di scorta*. Questa chiave, che doveva essere in possesso di un'agenzia del governo, consentiva di decriptare ogni comunicazione. L'FBI e il Ministero della giustizia volevano che questo chip venisse ampiamente usato nei dispositivi di comunicazione. Il "Clipper Chip" è stato fortemente avversato perché viola la privacy dei cittadini. Ogni tipo di protocollo crittografico può essere tecnicamente costruibile, un altro discorso è se sia politicamente accettabile o meno.

Le idee della crittografia hanno molte applicazioni oltre alla trasmissione di messaggi segreti. Per esempio è possibile verificare che i messaggi siano stati effettivamente spediti da chi dice di averlo fatto: questa è la "autenticazione", senza la quale il commercio elettronico non potrebbe esistere. Ci sono metodi per fare sì che le persone possano votare senza che il loro voto possa essere visto da altri, neanche da chi gestisce il sistema di elaborazione, ma riuscendo in ogni modo a scrutinare i voti

in modo corretto. È anche possibile giocare a carte al telefono, che può sembrare inutile e sciocco fino a quando non si comprende che stringere accordi commerciali non è molto diverso dal giocare a poker.

Ma come è mai possibile mischiare un mazzo di carte e giocare via telefono in competizione con una persona all'altro capo della comunicazione e di cui non ci si può fidare? Come si può riconoscere se qualcuno ha intercettato un messaggio, l'ha modificato e l'ha consegnato al posto dell'originale? Se non si potessero fare queste cose non sarebbe possibile fare commercio elettronico. Occorre impedire a criminali esperti dal punto di vista tecnico di poter forgiare false autorizzazioni di prelievo da conti bancari a partire dai dati intercettati sulla linea che connette il terminale del bancomat alla banca. Occorre che ditte malintenzionate non possano tecnicamente far fallire un concorrente generando falsi ordini o falsi contratti. Con la crittografia moderna questi miracoli possono essere compiuti e le seguenti attività indicano come.

Ci sono molti libri interessanti sui codici crittografici e sulla loro violazione. *Codebreakers: the inside story of Bletchley Park* curato da Hinsley e Stripp, fornisce una descrizione diretta di come alcuni dei primi computer vennero utilizzati per decodificare i codici militari durante la seconda guerra mondiale, consentendo così la fine delle ostilità e salvando in questo modo molte vite umane.

Attività 17

Condividere i segreti — *Protocolli che nascondono l'informazione*

Sommario

I protocolli crittografici ci consentono di condividere informazioni con altre persone, mantenendo nel contempo un sorprendente livello di privacy. Questa attività illustra una situazione nella quale l'informazione viene condivisa sebbene non venga rivelata: un gruppo di studenti calcolerà la media delle loro età senza che alcuno di loro debba rivelare la propria.

Abilità

- ✓ Calcolare una media
- ✓ Generare numeri casuali
- ✓ Lavorare in gruppo

Età

- ✓ a partire dai 7 anni

Materiale

Ogni gruppo deve avere:

- ✓ un blocchetto di carta
- ✓ una penna.

Condividere i segreti



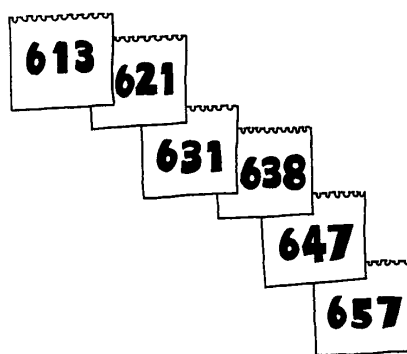
Introduzione

Scopo di questa attività è di calcolare la media delle età di un gruppo di studenti senza che ognuno di essi debba rivelare la propria. In alternativa con lo stesso metodo è possibile calcolare la media del reddito (della paghetta) degli studenti del gruppo o la media di qualche altro dato riservato. Questo metodo di calcolo funziona particolarmente bene con gli adulti, perché sono spesso particolarmente riservati su dati riservati come l'età o il reddito.

Il gruppo deve essere formato da almeno tre studenti.

Discussione

1. Spiegate al gruppo che volete conoscere l'età media dei partecipanti senza che ognuno debba dire agli altri la propria età. Chiedete suggerimenti su come pensano di fare e se pensano che questo problema ammetta una soluzione.
2. Selezionate da sei a dieci studenti (circa) per fare l'esperimento. Date il blocchetto al primo studente e chiedetegli di scrivere un numero casuale di tre cifre sul primo foglio. In questo esempio poniamo che lo studente abbia scelto 613.
3. A questo punto il primo studente deve strappare il primo foglietto dal blocco, aggiungere la propria età al numero casuale scelto e scrivere il risultato della somma nel secondo foglietto. Se lo studente ha otto anni il numero da scrivere nel secondo foglietto è 621. Lo studente deve tenere il foglio staccato dal blocco e non mostrarlo a nessuno.
4. Il blocchetto viene passato al secondo studente, che stacca dal blocco il foglio in cima, quello con il numero scritto dal collega,



calcola la somma fra tale numero e la propria età e scrive il totale sul foglietto successivo.

5. Occorre ora continuare il processo che vede ogni studente strappare un foglietto dal blocco, sommare la propria età e scrivere il risultato sul foglietto successivo. Il meccanismo continua fino a quando tutti gli studenti non hanno ricevuto il blocco.
6. A questo punto il blocco viene riconsegnato al primo studente che sottrae dal valore scritto sul blocco quello del numero casuale che aveva scelto all'inizio. Nell'esempio il blocco è stato fatto girare fra i cinque studenti del gruppo e al numero finale, 657, viene sottratto il numero casuale iniziale, 613. Il risultato è 44. Questa è la somma delle età degli studenti. La media a questo punto può essere calcolata in modo semplice dividendo questa somma per il numero degli studenti del gruppo. Nel nostro esempio l'età media risulta di 8.8 anni.
7. Sottolineate che, se ogni studente distrugge il foglietto che ha staccato dal blocchetto, nessuno può risalire all'età individuale di ognuno di loro, a meno che due studenti del gruppo non decidano di collaborare.

Variazioni ed estensioni

Il sistema può essere adattato per consentire votazioni segrete facendo sommare al risultato uno se il voto è sì e zero se è no. Ovviamente se non vengono rispettate le regole e qualcuno somma più di uno (o un numero negativo) la votazione non è equa. Tuttavia questa frode corre facilmente il rischio di essere rilevata se la somma totale dei voti si diviene maggiore del numero delle persone che hanno votato.

Cosa c'entra tutto questo?

I computer memorizzano tante informazioni personali: il saldo del nostro conto corrente, le nostre reti sociali, quanto dobbiamo pagare di tasse, da quanto tempo abbiamo la patente di guida, la nostra storia creditizia, i risultati degli esami, la nostra cartella clinica, ecc. La privacy è molto importante! Ma dobbiamo al tempo stesso avere la possibilità di condividere alcune di queste informazioni con gli altri. Per esempio quando paghiamo col Bancomat il conto alla cassa consentiamo al gestore del negozio di verificare che c'è denaro a sufficienza nel nostro conto corrente per pagare l'importo richiesto.

Spesso finiamo con fornire più informazioni del necessario. Per esempio, se facciamo una transazione elettronica in un negozio riveliamo anche quale sia la nostra banca, il nostro nome e se paghiamo con un assegno anche il numero del conto corrente. Inoltre la banca scopre in quale negozio abbiamo deciso di fare shopping. Le banche possono creare un profilo del cliente analizzando ciò che acquista, quanto il cliente spende in media e quali sono i luoghi che normalmente visita. Se avessimo pagato con il denaro contante nessuna di queste informazioni sarebbe stata rivelata. Molte persone non si preoccupano troppo delle informazioni che possono essere carpite in questo modo, ma c'è la possibilità che queste informazioni vengano usate per compiere abusi o per spedire pubblicità selettiva (per esempio inviando le offerte delle agenzie di viaggio a chi spende tanto in biglietti aerei) o per discriminare (per esempio fornendo servizi migliori a chi ha la carta riservata ai clienti più ricchi della banca) oppure anche per ricattare (minacciando di rivelare i dettagli di pagamenti che possono creare imbarazzo). In ogni modo, le persone potrebbero anche semplicemente cambiare le loro abitudini di acquisto se pensassero che c'è chi le sta spiando.

Questa perdita di privacy è comunemente accettata dalla maggior parte della popolazione, anche se esistono protocolli crittografici che ci consentirebbero di fare transazioni finanziarie con lo stesso livello di privacy del denaro contante. Potrebbe sembrare difficile da credere che il denaro possa essere spostato dal vostro conto corrente al conto del negozio senza che nessuno sappia nè da dove il denaro venga nè dove stia andando. Questa attività mostra che è plausibile che esista una soluzione a questo problema: entrambe le situazioni richiedono una condivisione limitata di informazioni e nell'esempio precedente abbiamo visto che ciò è possibile a patto di scegliere un protocollo appropriato.

Per ulteriori approfondimenti

Un articolo famoso che mette in evidenza gli argomenti qui trattati è stato scritto da David Chaum, con il titolo provocatorio "Security without

identification: transaction systems to make Big Brother obsolete.”
(sicurezza senza identificazione: sistemi di transazioni e che rendono
obsoleto il grande fratello) [2]. L'articolo, di lettura non eccessivamente
impegnativa, fornisce esempi di semplici di protocolli che nascondono
le informazioni. Mostra anche come transazioni totalmente private
possano essere fatte usando “denaro elettronico”. L'articolo compare
nel numero di ottobre 1985 della rivista *Communications of the ACM*.

Attività 18

Testa o croce in Perù — *I protocolli crittografici*

Sommario

Questa attività mostra come si possa compiere un'azione semplice ma che sembra impossibile: fare una scelta casuale, come in una partita a testa o croce, fra persone collegate da una linea telefonica e che non necessariamente si fidino l'una dell'altra.

Conoscenze richieste

- ✓ Logica booleana
- ✓ Funzioni
- ✓ Enigmistica

Età

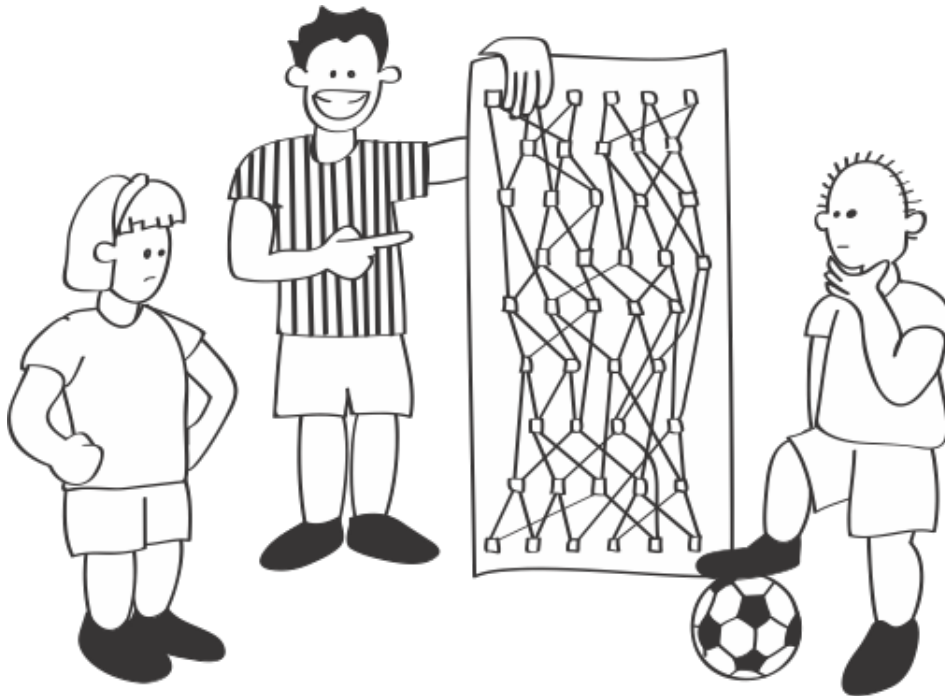
- ✓ A partire dai 9 anni

Materiale

Ogni gruppo di studenti deve avere:

- ✓ una copia del foglio di lavoro: Testa o croce in Perù;
- ✓ due dozzine circa di piccoli bottoni o gettoni di due diversi colori.

Testa o croce in Perù



Introduzione

Questa attività è stata inizialmente ideata quando uno degli autori (MRF) stava lavorando in Perù. Questo è il motivo del titolo. È possibile adattare la storia usando nomi e situazioni locali.

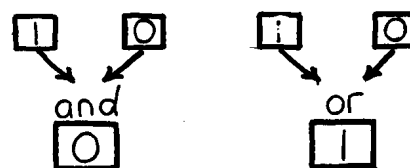
La squadre di calcio di Lima e Cuzco devono decidere chi giocherà in casa per la prossima partita di campionato. Il modo più semplice è di tirare una moneta e giocarsi la scelta a testa o croce. Ma le due città sono distanti e nè Alicia, che rappresenta il Lima, nè Basilio che rappresenta il Cuzco, vogliono spendere il tempo e il denaro per incontrarsi solo per tirare la moneta. Possono prendere la decisione parlando al telefono? Alicia potrebbe tirare la moneta e Basilio decidere se vuole testa o croce. Ma non funzionerebbe perché se Basilio decide "testa" Alicia può semplicemente dire "mi dispiace è venuto croce" senza che Basilio possa controllare. Alicia non è normalmente disonesta ma la partita è importante e le tentazioni sono forti. Anche posto che agisca in modo corretto, se Basilio perde sarà sicuro dell'onestà di Alicia?

Gli studenti saranno in grado di capire meglio questa unità se hanno già imparato la rappresentazione tramite numeri binari (attività 1, conta i punti), il concetto di parità (attività 4, la magia delle carte girate) e se hanno visto esempi di funzioni a senso unico nell'attività 15, la città turistica.

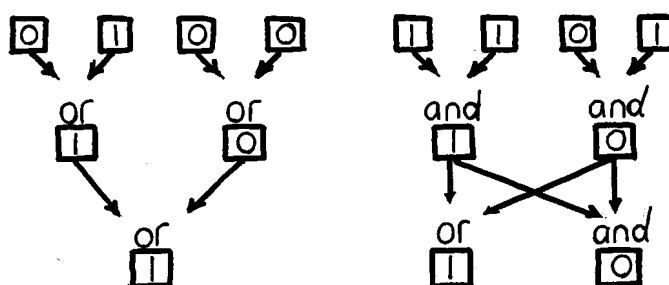
Ecco come Alicia e Basilio decidono di fare. Lavorando insieme, disegnano un circuito fatto di porte-`and` e porte-`or`, come spiegato in seguito. In linea di principio tutto questo può essere fatto al telefono anche se alla fine risulterebbe non poco noioso (si può pensare di usare l'e-mail o il fax). Durante la costruzione entrambi hanno l'interesse a fare in modo che il circuito sia sufficientemente complesso in modo che l'altro non possa barare. Il circuito finale viene pubblicato in modo che tutti lo possano vedere.

Le regole delle porte

`and` e `or` sono semplici. Ogni porta ha due entrate (input) e un'uscita (output). Ogni segnale fornito in ingresso assume uno dei due valori, 0 oppure 1, che possono essere interpretati rispettivamente come falso (false) e vero (true). L'uscita di una porta `and` vale 1 (vero/true) solo quando entrambe le entrate valgono 1 (vero/true). Di conseguenza l'uscita varrà 0 (falso/false) in tutti gli altri casi. L'output di una porta `or` vale 0 (falso/false) solo se entrambi i dati in ingresso valgono 0 (falso/false); se uno o entrambi gli ingressi valgono 1 (vero/true) allora il risultato sarà 1 (vero/true).

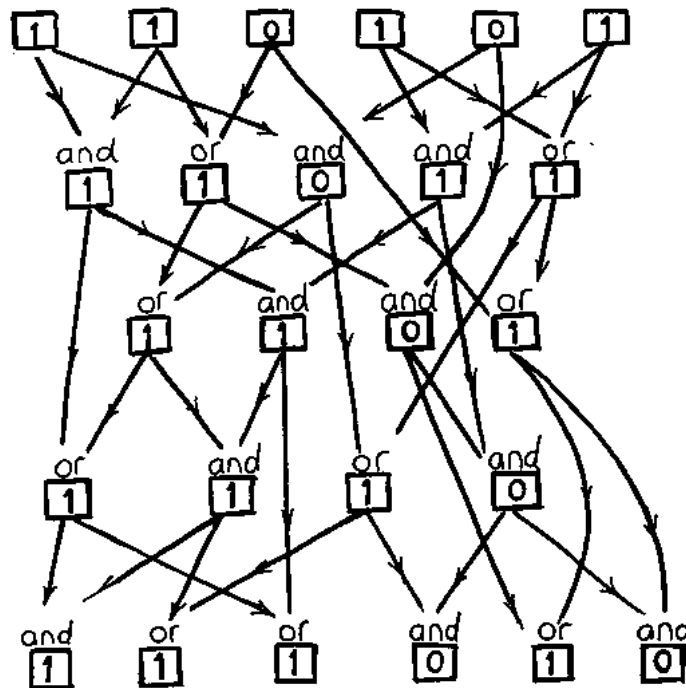


L'uscita di una porta può essere collegata all'ingresso di una porta (o a più porte) per produrre un effetto più complesso. Per esempio nella figura alla sinistra



l'uscita delle due porte `or` diventano dati in ingresso per una terza porta `or`. Nel circuito alla destra le uscite di entrambe le porte `and` in alto alimentano l'input delle due porte in basso: l'intero circuito ha quindi quattro ingressi e due uscite.

Per il gioco del "testa o croce in Perù" abbiamo necessità di un circuito ancora più complesso. Il circuito disegnato nel foglio di lavoro ha sei ingressi e sei uscite. Riportiamo qui di seguito come esempio i calcoli relativi ad una specifica configurazione dei valori in ingresso.



Per giocare a testa o croce al telefono tramite questo circuito si fa così: Alicia sceglie a caso un numero di sei bit, lo annota e lo tiene segreto. Mette i sei bit come valori di ingresso nel circuito e spedisce a Basilio i sei valori di uscita. Quando Basilio riceve questi valori tenta di indovinare la parità del numero segreto di Alicia. Se il circuito è sufficientemente complesso Basilio non potrà indovinare quale sia il numero segreto di Alicia e quindi la scelta dovrà essere casuale (potrebbe giocare da solo a testa o croce per determinare la scelta). Basilio vince e quindi la partita si svolgerà a Cuzco se la sua risposta si rivelerà corretta. Alicia vince se la scelta di Basilio si rivelerà errata. Quando Basilio ha detto ad Alicia la sua scelta, Alicia può rivelare il suo numero segreto così che Basilio possa controllare e confermare la correttezza del risultato ottenuto.

1. Dividete gli studenti in piccoli gruppi, date ad ogni gruppo il circuito e alcuni gettoni e spiegate loro la storia. La situazione sarà probabilmente più convincente se gli studenti immagineranno i capitani delle loro squadre giocare a testa o croce coi capitani delle squadre delle scuole rivali. Stabilite una convenzione per i colori dei gettoni (per esempio il rosso è 0 e il blu è 1) e fate contrassegnare o colorare la legenda in alto.
2. Mostrate agli studenti come mettere i gettoni sugli input in modo da mostrare i numeri che Alicia sceglie. Quindi spiegate le regole delle porte `and` e `or` che sono riportate in fondo al foglio (valutate l'idea di far colorare i valori dei bit in modo coerente col colore dei gettoni).
3. Mostrate come funziona il circuito, mettendo gettoni nei nodi in

modo da poter derivare le uscite delle porte corrispondenti. Questa operazione deve essere fatta con accuratezza; la tavola seguente (che non deve essere data agli studenti) mostra gli output per ogni possibile configurazione di input. Serve per vostra referenza, da consultare in caso di dubbio.

Input	000000	000001	000010	000011	000100	000101	000110	000111
Output	000000	010010	000000	010010	010010	010010	010010	010010
Input	001000	001001	001010	001011	001100	001101	001110	001111
Output	001010	011010	001010	011010	011010	011010	011010	011111
Input	010000	010001	010010	010011	010100	010101	010110	010111
Output	001000	011010	001010	011010	011010	011010	011010	011111
Input	011000	011001	011010	011011	011100	011101	011110	011111
Output	001010	011010	001010	011010	011010	011010	011010	011111
Input	100000	100001	100010	100011	100100	100101	100110	100111
Output	000000	010010	011000	011010	010010	010010	011010	011010
Input	101000	101001	101010	101011	101100	101101	101110	101111
Output	001010	011010	011010	011010	011010	011010	011010	011111
Input	110000	110001	110010	110011	110100	110101	110110	110111
Output	001000	011010	011010	011010	011010	111010	011010	111111
Input	111000	111001	111010	111011	111100	111101	111110	111111
Output	001010	011010	011010	011010	011010	111010	011010	111111

4. Ora ogni gruppo deve eleggere una Alicia e un Basilio. Alternativamente, il gruppo si può spezzare in due parti e ogni sottogruppo fare le veci di Alicia o di Basilio. Alicia deve scegliere un numero random da mettere in ingresso nel circuito, deve quindi calcolare i valori di uscita (output) e comunicarli a Basilio. Basilio deve indovinare la parità del numero usato da Alicia come ingresso del circuito (cioè se ha un numero pari o dispari di valori posti a uno/vero/true). Sarà chiaro che la decisione di Basilio sarà casuale. Alicia a questo punto può mostrare il numero mantenuto segreto e Basilio vincerà se avrà indovinato la parità corretta. A questo punto Basilio può controllare che Alicia non abbia modificato il numero: sarà sufficiente controllare con il circuito che fornisca lo stesso output che Alicia aveva comunicato.

A questo punto la partita di “testa o croce” è terminata.

Basilio potrebbe barare se riuscisse a indovinare il numero di sei bit usato da Alicia per generare l'output ricevuto. L'interesse di Alicia è quindi di usare una funzione a senso unico (come discusso nell'attività 15) per evitare che Basilio possa barare. Una funzione a senso univo (one-way) è facile da calcolare ma è difficile sapere quale valore di ingresso abbia determinato uno specifico valore di uscita.

Alicia potrebbe barare se riuscisse a scoprire due valori di diversa

parità che producono esattamente gli stessi valori in uscita. Così facendo, qualsiasi fosse la scelta di Basilio, Alicia potrebbe sostenere che ha sbagliato ad indovinare la parità. L'interesse di Basilio è di assicurare che il circuito non faccia corrispondere configurazioni di ingresso differenti agli stessi valori in uscita o almeno che sia difficile trovare queste corrispondenze.

5. Provate a vedere se gli studenti riescono a trovare casi nei quali Alicia o Basilio possano barare. Per esempio nella prima riga della tabella si vede che molteplici configurazioni di input generano in output 010010, per esempio 000001, 000011, 000101, etc. Quindi se Alicia dichiara a Basilio il numero in output 010010 può poi scegliere 000001 se Basilio dice "pari" e 000011 se dice "dispari".

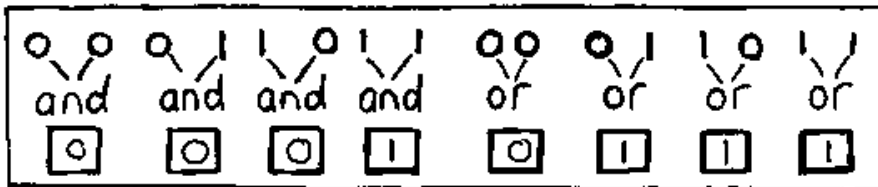
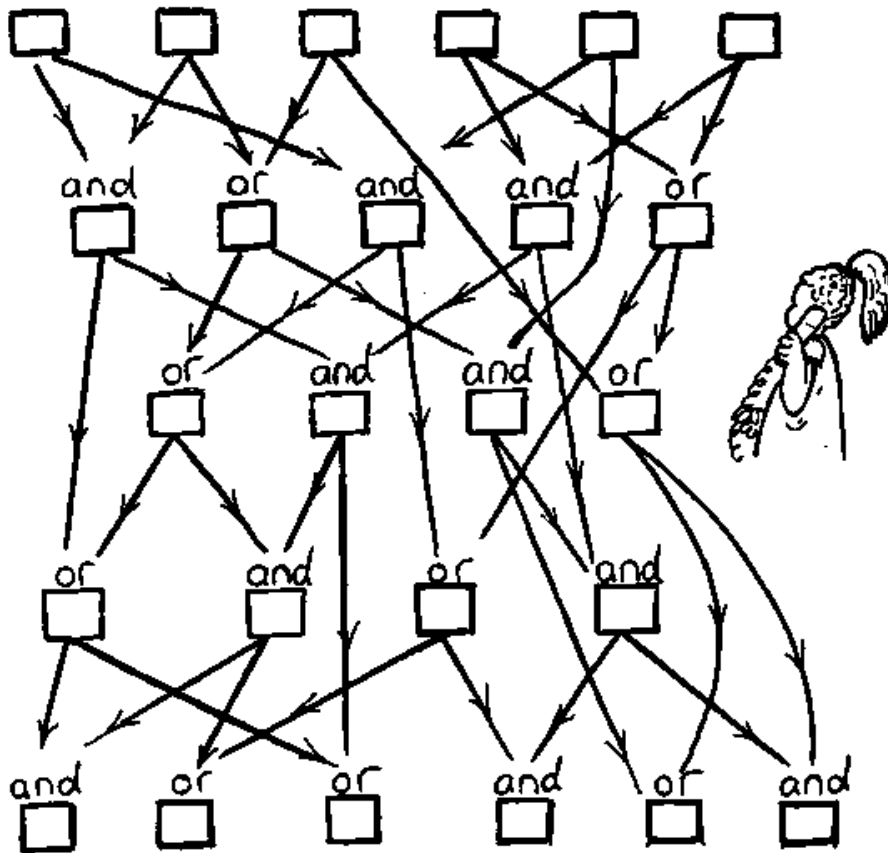
Con questo circuito è difficile barare per Basilio. Ma se per esempio l'output è 011000, allora l'input di Alicia non può che essere 100010, non c'è nessun'altra possibilità (potete controllare nella tabella). Quindi se questo è il numero che Alicia ha dato a Basilio, quest'ultimo può scegliere la parità pari ed è sicuro di vincere. Nella realtà se si usasse un computer per risolvere questo problema i numeri in input e in output avrebbero un numero di bit molto maggiore e quindi ci sarebbero troppe configurazioni da provare (a ogni bit aggiunto raddoppia il numero delle configurazioni) per trovare un caso simile a questo.

6. Ora chiedete ai gruppi di studenti di ideare i loro circuiti per questo gioco. Guardate se riescono a creare circuiti per i quali sia facile barare per Alicia e altri nei quali sia facile barare per Basilio. Nulla vieta di avere un numero di bit in ingresso e in uscita diversi fra loro e diversi da sei.

Foglio di lavoro: testa o croce in Perù



KEY □ = 1 = true
 □ = ● = false

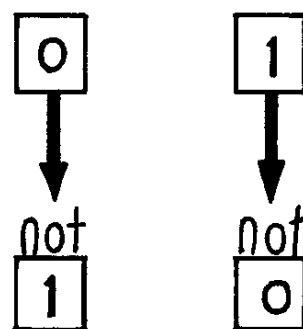


Variazioni ed estensioni

1. Un chiaro problema pratico è la cooperazione necessaria per costruire un circuito che sia accettabile sia per Alicia sia per Basilio. Questa può essere una fase divertente per gli studenti, ma può rendere la procedura inapplicabile in pratica, in modo particolare se gli accordi devono essere presi per telefono. In ogni caso c'è una semplice alternativa: sia Alicia sia Basilio costruiscono un circuito ed entrambi rendono pubblico lo schema del proprio circuito. Alicia fornisce il proprio input ad entrambi i circuiti e combina le due sequenze di output in questo modo: il valore finale sarà uno se i bit corrispondenti hanno lo stesso valore, zero in caso contrario. In questo modo nessuno dei due partecipanti può barare (a meno che non barino entrambi) se anche una delle due funzioni non fosse a senso unico, la combinazione delle due lo sarebbe.

Le prossime due variazioni non sono relative ai protocolli crittografici o al problema di giocare a testa o croce, bensì all'idea del circuito costruito mediante le porte logiche `and` e `or`. Verranno esplorate alcune importanti nozioni non solo relative ai circuiti per l'elaborazione ma alla logica stessa. Questo tipo di logica viene chiamata *Algebra di Boole* dal nome del matematico George Boole (1815-64).

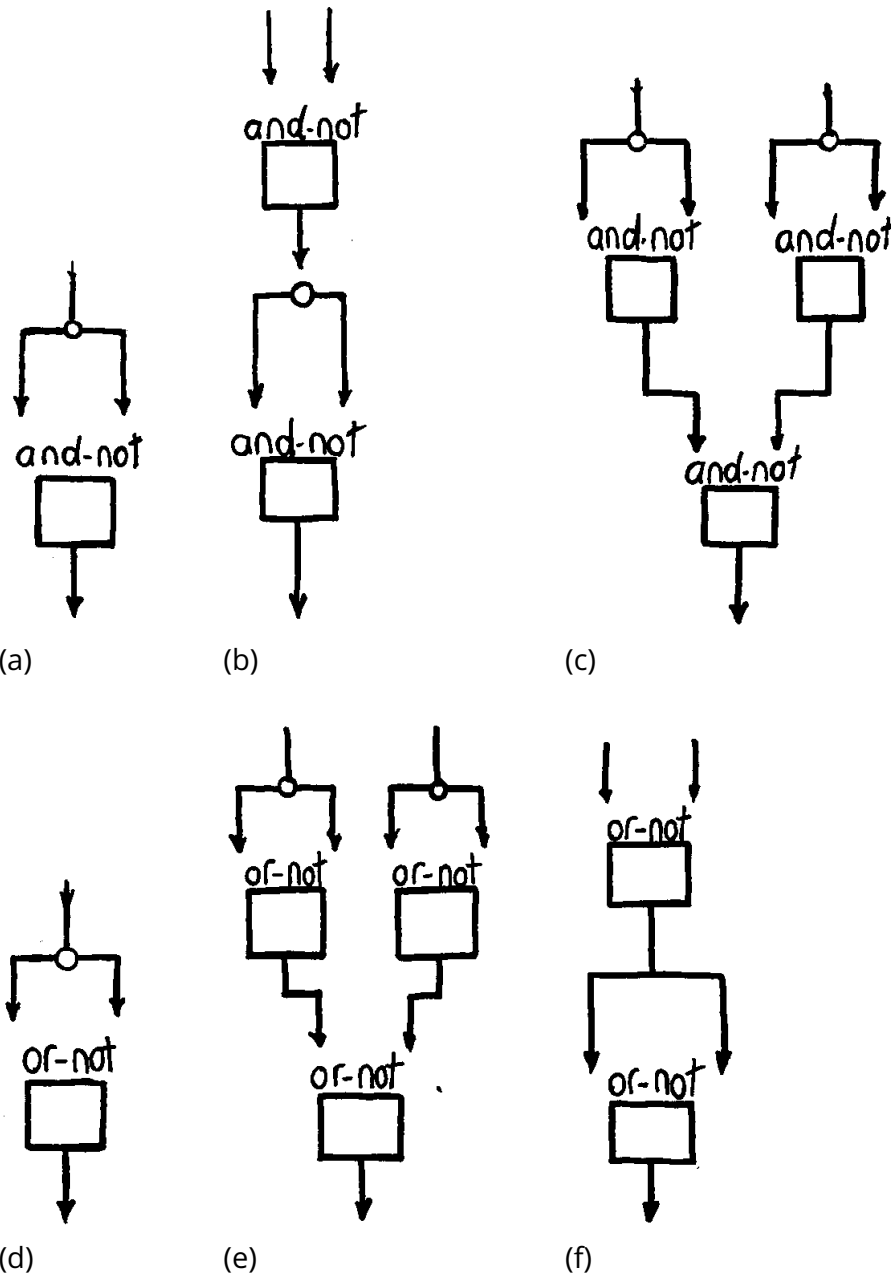
2. Gli studenti potranno aver notato che la configurazione formata solo da input di valore zero, 000000, produce in output sempre 000000 e, in maniera duale, la configurazione con tutti gli input uguali ad uno, 111111, produce in output 111111 (ci possono essere altre configurazioni di input che forniscono questi output, per esempio 000010 produce 000000 mentre 110111 produce 111111). Questa proprietà deriva dal fatto che il nostro circuito usa solamente porte `and` e `or`. Se si aggiunge un nodo `not` che prende in input un solo bit e ne inverte il valore in uscita ($0 \rightarrow 1$, $1 \rightarrow 0$), gli studenti possono costruire circuiti che non hanno questa proprietà.



3. Altri due circuiti importanti sono gli `and-not` e gli `or-not` (normalmente abbreviati rispettivamente coi nomi di `nand` e `nor`) che sono come `and` e `or` ma sono seguiti da un `not`. Quindi $a \text{ nand } b = \text{not}(a \text{ and } b)$. L'introduzione di questi circuiti non permette di costruire nuovi tipi di circuito, più di quelli realizzabili con le porte che abbiamo già studiato, perché il loro effetto potrebbe essere sempre ricostruito usando porte `and`, `or` e `not`. In ogni caso, questi circuiti hanno una importante proprietà: usando solamente porte di tipo `nand` (o similmente solo porte di tipo `nor`) è possibile realizzare tutti gli altri circuiti.

Avendo introdotto le porte `nand` e `nor`, una sfida per gli studenti è quella di trovare il modo di realizzare tutte le porte mediante altre e arrivare a realizzare ogni tipo di porta connettendo più porte dello stesso tipo.

Le porte `not`, `and` e `or` possono essere costruite a partire dalle porte `nand` (in alto) o con sole porte `nor` (in basso).



Cosa c'entra tutto questo?

Negli ultimi anni abbiamo assistito ad un enorme aumento del commercio elettronico ed è essenziale garantire uno scambio sicuro del denaro, delle transazioni riservate nonché la gestione di documenti firmati e con valore legale. Lo scopo della crittografia è di consentire una comunicazione sicura e riservata. Alcuni decenni fa i ricercatori scoprirono un risultato controintuitivo: la segretezza può essere garantita anche se certe informazioni sono pubbliche. Il risultato è la "crittografia a chiave pubblica" descritta nell'attività 19, Kid Krypto, che è oggi ampiamente usata come strumento principale per lo scambio sicuro di informazioni. Per esempio potete aver impostazioni nel vostro browser web per attivare SSL (Secure Sockets Layer) o anche TLS (Transport Layer Security); questi sistemi sono basati sulla crittografia a chiave pubblica e abilitano il vostro browser a creare connessioni sicure con certi siti web quali la vostra banca. La comunicazione in questi casi sarebbe sicura anche se ci fosse qualcuno che sta intercettando il vostro traffico sulla rete.

La crittografia non serve solo per mantenere le cose segrete, ma anche per porre limiti a ciò che gli altri possono scoprire e anche per creare "affidabilità" fra persone che sono geograficamente distanti. Le regole formali (dette "protocolli") per le transazioni crittografiche sono state ideate per permettere questi apparentemente impossibili servizi, come le firme digitali non falsificabili o l'abilità di possedere un segreto (come una password) e farne uso senza rivelare quale esso sia. Giocare a testa o croce al telefono è un problema analogo sebbene più semplice. Anch'esso sembra essere impossibile, ma come abbiamo visto non è così.

Nelle situazioni reali Alicia e Basilio non disegnerebbero un circuito per conto proprio ma userebbero un programma capace di costruire "logicamente" il circuito. Probabilmente nessuno dei due sarebbe interessato ai dettagli costruttivi del programma, ma entrambi vorrebbero essere sicuri che l'altro non possa influenzare l'esito della decisione indipendentemente da quanto sia esperto con l'informatica o da quanto tenacemente ci provi.

In linea di principio, ogni disputa potrebbe essere risolta facendo affidamento ad un giudice terzo imparziale. Il giudice potrebbe ricevere il circuito, il numero random usato da Alicia, il messaggio mandato a Basilio con l'output del circuito e la scelta fatta da Basilio rispedita come risposta ad Alicia. Alla fine della transazione tutte queste sono informazioni pubbliche quindi tutti i partecipanti dovranno concordare sui dati che hanno contribuito al risultato. Il giudice potrà immettere il numero di Alicia nel circuito e controllare che l'output sia quello spedito a Basilio e quindi stabilire se la decisione sia stata presa in modo

corretto o meno. È forse superfluo dire che il fatto importante è che esiste una chiara procedura di verifica della corretta applicazione delle regole ed è quindi molto improbabile che una disputa possa mai nascere. Se invece Alicia e Basilio avessero giocato a testa o croce con una reale moneta, nessun giudice sarebbe potuto intervenire in caso di disputa.



Un circuito con pochi componenti come quello illustrato in questa attività non potrebbe essere applicato a casi reali perché è possibile creare una tabella che comprenda tutti i casi possibili e usarla per barare. Usando trentadue bit come input si potrebbe avere una maggior protezione. Questo però non potrebbe garantire che sia difficile barare, dipende da come è fatto il circuito. Altri metodi possono essere basati sulla funzione a senso unico definita nell'attività 15, la città turistica. I metodi usati in pratica sono basati sulla "fattorizzazione di numeri molto grandi", che è nota come un problema molto complesso

(anche se, come vedremo alla fine della prossima attività, non è *NP-completo*). È facile controllare se un numero è un fattore di un altro, ma trovare i fattori di un numero è un'attività che richiede tantissimo tempo. Questo rende molto complesso per Alicia e Basilio (e per il giudice) poter fare i calcoli a mano quindi, come notato sopra, tutti i calcoli vengono svolti da programmi di uso comune.

Le firme digitali sono basate su una simile idea. Alicia, rendendo pubblico l'output del circuito per uno specifico input segreto che lei ha scelto, può provare in un secondo tempo che è stata lei a generare quell'output. Con una reale funzione a senso unico nessun altro può produrre un input capace di generare lo stesso messaggio. Nessun impostore può farsi passare per Alicia. Per fare una reale firma digitale è necessario un protocollo più complesso per garantire che Alicia possa firmare un messaggio e anche per garantire che gli altri possano controllare l'autenticità della firma di Alicia anche nel caso che Alicia stessa sostenga di non averlo firmato. Il principio è in ogni caso lo stesso.



Un'altra applicazione consente di giocare a poker al telefono, senza alcun arbitro che dia le carte e registrando tutte le *mani* di entrambi i giocatori. Tutto deve essere fatto dai giocatori stessi, con il ricorso ad un giudice alla fine del gioco solo in caso di disputa. Situazioni simili accadono seriamente nelle contrattazioni commerciali. Ovviamente i giocatori devono tenere le loro carte segrete durante il gioco. Ma devono anche essere onesti, non possono sostenere di avere un asso se non l'hanno veramente. Questo può essere controllato attendendo la

fine del gioco e consentendo poi ad ogni giocatore di guardare la prima mano di tutti gli altri e la sequenza di azioni fatte. Un altro problema è come fare a tenere tutte le carte di ogni giocatore segrete fino alla fine del gioco. Sorprendentemente è possibile consentire questo usando un protocollo crittografico non molto diverso dal gioco a testa o croce qui descritto.

I protocolli crittografici sono estremamente importanti nelle transazioni elettroniche, per identificare il titolare di una carta di credito, per autorizzare l'uso di un telefono cellulare, o per autenticare il mittente di un messaggio di e-mail. L'abilità di fare queste azioni in modo affidabile è cruciale per il successo del commercio elettronico.

Per ulteriori approfondimenti

Il libro di Harel con titolo *Algorithmics* [5] discute le firme digitali e i protocolli crittografici correlati. Mostra anche come giocare a poker al telefono, una idea introdotta per la prima volta nel 1981 in un capitolo dal titolo "mental poker" del libro *The Mathematical Gardener* curato a D.A. Klarner [8]. *Cryptography and data security* di Dorothy Denning [11] è un testo scientifico eccellente sulla crittografia. *Turing Omnibus* di Dewdney [3] ha una sezione dedicata alla logica di Boole che discute i blocchi usati per la costruzione del circuito presentato in questa attività.

Attività 19

Kid Krypto — *La crittografia a chiave pubblica*

Sommario

La crittografia è il concetto chiave della sicurezza delle informazioni. E il concetto fondamentale della crittografia moderna è che usando solo informazioni pubbliche un mittente può chiudere a chiave il messaggio in uno scrigno (logico) che solo il legittimo destinatario potrà (privatamente) aprire. È come se ogni studente comprasse un lucchetto, ci scrivesse sopra il proprio nome e lo lasciasse aperto sul tavolo tenendo la chiave. I lucchetti ai quali facciamo riferimento sono del tipo che si usa normalmente per la catena della bicicletta o per chiudere l'armadietto della piscina, non hanno necessità della chiave per venir chiusi, basta stringere il ferro a U fino a che non fa click. Se ora voglio spedire un messaggio a uno studente, cerco sul tavolo il lucchetto con il suo nome e uso il lucchetto per chiudere una scatola nella quale avrò messo il messaggio. Anche se la scatola cadesse nelle mani sbagliate nessuno potrà aprire la scatola se non il destinatario legittimo che ha la chiave del lucchetto. Con questo schema non è necessaria alcuna precedente forma di comunicazione, non servono chiavi segrete.

Questa attività mostra come questo può essere fatto in modo digitale. E nel mondo digitale non è necessario prendere il lucchetto dal tavolo, i lucchetti logici possono essere copiati. Se dovessimo costruire una copia di un lucchetto fisico, dovremmo smontarlo e inevitabilmente scopriremmo come funziona a magari potremmo forgiare una chiave falsa. Ma nel mondo digitale è possibile consentire alle persone di copiare i lucchetti senza che possano scoprire la chiave! Sembra impossibile? Continuate a leggere.

Abilità

- ✓ Saper risolvere giochi enigmistici

Età

- ✓ A partire da 11 anni.

Materiale

Create gruppi di circa quattro studenti. In ogni gruppo individuate due sottogruppi. A ogni sottogruppo viene data una copia del foglio di lavoro *Mappe Kid Krypto*. Quindi ogni gruppo ha necessità di

- ✓ due copie delle Mappe Kid Krypto (pag. 235)

Servirà anche:

- ✓ uno strumento per mostrare alla classe lo schema Codifica Kid Krypto (pag. 236)
- ✓ uno strumento per disegnare note sullo stesso schema.



Kid Krypto

Introduzione

Questa è l'attività più impegnativa dell'intero libro dal punto di vista tecnico. Anche se può essere gratificante, richiede tanta attenzione e tanta concentrazione per avere successo. Gli studenti dovrebbero aver già studiato gli esempi di funzioni a senso unico dell'attività 15, la città turistica ed è di aiuto se hanno già completato le altre attività di questa sezione (l'attività 17, condividere i segreti e l'attività 18, la moneta peruviana. Questa attività usa anche idee introdotte nelle attività 1, conta i punti e 5, indovina indovinello).

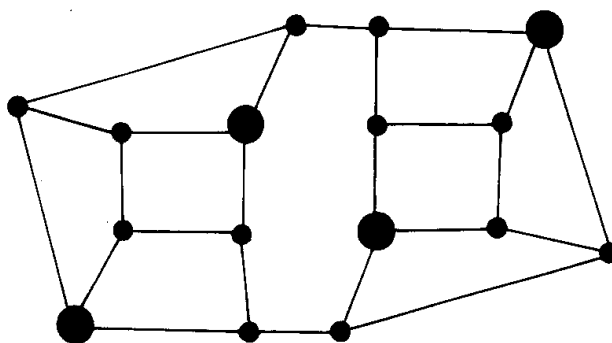
Anna deve mandare a Bruno un messaggio segreto. Normalmente noi pensiamo che il messaggio segreto sia una frase o un breve testo. Nell'esercizio seguente Anna spedisce un carattere o meglio un numero che rappresenta un carattere. Anche se questo messaggio può sembrare semplicistico, tenete a mente che Anna può spedire un'intera sequenza di questi "messaggi" per formare una frase e che questo lavoro verrà fatto in realtà da un computer. Vedremo come nascondere il numero di Anna in un messaggio criptato con la chiave di Bruno in modo tale che se anche qualcuno intercettasse il messaggio non sarebbe in grado di decodificarlo. Solo Bruno, che ha la chiave del lucchetto, potrà leggerlo.

Useremo delle mappe per nascondere il messaggio. Non sono le mappe del tesoro, dove una X indica il punto esatto dove scavare, ma piuttosto mappe stradali come quelle della città turistica 15, dove le linee sono le strade e i punti gli incroci. Ogni mappa ha una versione pubblica, cioè il lucchetto, e una versione privata, cioè la chiave.

Discussione

Nel foglio di lavoro Codifica Kid Krypto (pag. 236) c'è la chiave pubblica di Bruno. Non è un segreto, Bruno può mettere questa mappa sul tavolo (o su una pagina web) perché tutti possano vederla oppure (equivalentemente)

può darne una copia a chiunque voglia spedirgli un messaggio. Anna ha una copia della mappa pubblica di Bruno così come tutti gli altri. La figura qui a destra mostra la chiave privata di Bruno. È uguale alla sua chiave pubblica eccetto che per il fatto che alcuni degli incroci sono stati

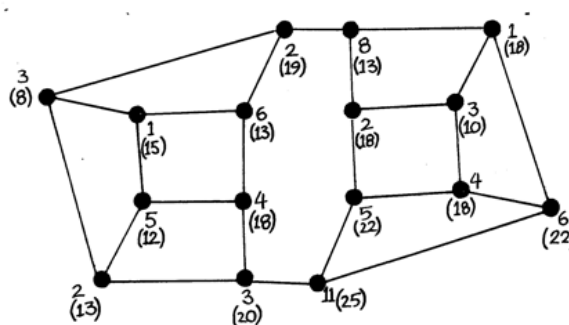


evidenziati disegnando un punto di dimensione maggiore. Bruno deve mantenere segreta questa mappa.

Questa attività è bene farla tutti insieme in classe, almeno all'inizio, perché richiede una notevole quantità di lavoro. Anche se le operazioni non sono particolarmente difficili occorre svolgerle con grande accuratezza. Ogni errore può far fallire l'esperimento. È importante che gli studenti si rendano conto di come sia sorprendente che questo tipo di crittografia possa esistere, sembra infatti impossibile perché questo stupore fornisce la forza di arrivare al risultato nonostante lo sforzo richiesto. Un elemento che fornisce molta motivazione agli studenti è che questo metodo può consentire loro di scambiare messaggi segreti all'interno della classe e anche se l'insegnante sa come il messaggio è codificato, non sarà in grado di decodificarlo.

1. Mostrate

la mappa pubblica di Bruno (il foglio di lavoro *codifica kid crypto* a pag. 236). Decidete quale numero Anna voglia inviare. A questo punto scrivete numeri casuali in ogni incrocio nella mappa in modo che la somma di tutti i



numeri sia il numero da trasmettere (sono i numeri indicati senza parentesi nella figura). Anna nell'esempio ha scelto di inviare 66, quindi la somma di tutti i numeri senza parentesi è 66. Potete usare anche numeri negativi, se volete, per ottenere il valore cercato.

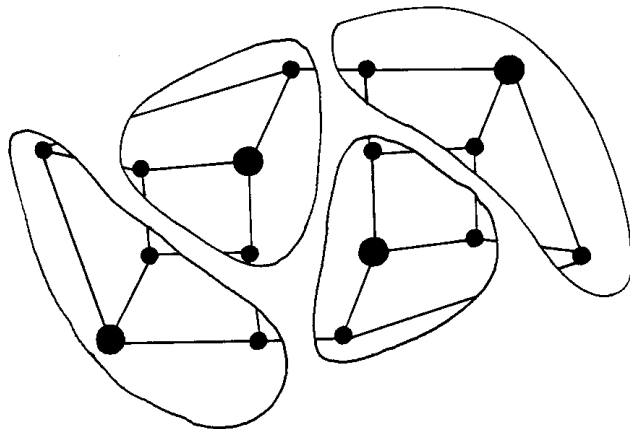
2. Ora Anna deve calcolare i valori da mandare a Bruno. Se spedisse i numeri scritti sulla mappa chiunque leggendo il messaggio potrebbe sommare i numeri assieme ed ottenere 66, il messaggio segreto. Non saranno questi i numeri inviati a Bruno.

Invece, per ogni incrocio considerate l'incrocio e tutti gli incroci "vicini", quelli che si raggiungono percorrendo una sola strada, e sommate i valori ottenuti (quelli cioè nell'incrocio stesso e in tutti i vicini). Scrivete questi numeri fra parentesi, come in figura, o usando una penna di diverso colore. Per esempio se esaminiamo l'incrocio (cioè il nodo) più a destra nella figura vediamo che ha il valore 6 e ha tre incroci vicini che hanno valore 1, 4 e 11. Il totale è quindi 22. Ora ripetete questo procedimento per tutti gli incroci della mappa: dovrete ottenere tutti i numeri indicati fra parentesi nella figura.

3. Anna spedirà a Bruno la mappa con solamente i numeri fra parentesi. Cancellate quindi i numeri senza parentesi (e le note dei

calcoli, se ci sono!) lasciando solo i numeri da inviare (o prendete una nuova mappa e ricopiate solo questi numeri). Provate a vedere se qualche studente è in grado di dire quale fosse il numero originale del messaggio. Nessuno sarà in grado.

4. Solo chi è in possesso della chiave privata di Bruno può decodificare il messaggio e trovare il numero che Anna ha inviato. Per decodificare il messaggio è sufficiente che Bruno prenda la mappa segreta e sommi fra loro i numeri presenti nei punti evidenziati (quelli più grandi). Nell'esempio questi incroci contengono i valori 13, 13, 22 e 18 che sommati insieme danno proprio 66, il numero spedito da Anna.



5. Ma come funziona? La mappa è speciale. Supponete che Bruno prenda la sua mappa e tracci una linea attorno ad ogni incrocio evidenziato comprendente anche gli incroci "vicini". Provate a disegnare anche voi queste linee come nella figura a lato. Queste linee partizionate i nodi in modo che tutti gli incroci apparterranno a un pezzo (ed a uno solo, mai a due). Quindi sommare insieme i numeri di questi quattro incroci equivale a sommare tutti i numeri senza parentesi della prima mappa di Anna e quindi il risultato sarà il messaggio segreto.

Uffa! È stato necessario un sacco di lavoro per spedire una sola lettera. E se è necessario così tanto lavoro per una lettera la crittografia è veramente faticosa. Ma guardate ciò che avete ottenuto: completa segretezza usando una chiave pubblica, senza che i partecipanti abbiano dovuto mettersi d'accordo in anticipo per una chiave segreta. Potete pubblicare la vostra chiave pubblica in una bacheca e chiunque può usarla per spedirvi messaggi segreti, che nessuno è in grado di leggere se non conosce la chiave segreta. E nella vita reale tutti i calcoli vengono svolti da un programma, molto spesso da un modulo del vostro programma browser per il web, quindi è solo il vostro computer che deve fare il duro lavoro di calcolo.

Forse la vostra classe sarà felice di sapere che siete riusciti ad entrare nel club esclusivo di quanti hanno effettivamente provato un algoritmo di crittografia facendo i calcoli a mano. Normalmente gli studiosi lo considerano un compito quasi impossibile e poche persone lo hanno fatto.

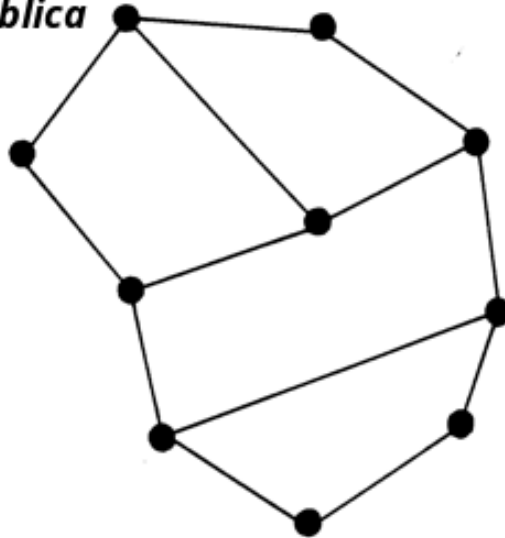
E cosa dire delle intercettazioni? La mappa usata da Bruno è simile a quella dell'attività della città turistica (attività 15), dove gli incroci evidenziati sono quelli che consentono di avere il furgoncino dei gelati ad un isolato di distanza al massimo. Se è facile per Bruno calcolare una nuova mappa partendo dai pezzi indicati nella mappa privata, è molto difficile per gli altri trovare il modo ottimale di porre i furgoncini a partire dalla mappa pubblica, a meno di non usare il metodo di *forza bruta* provando cioè ogni possibile scelta, prima provando con un furgoncino, poi con due e così via fino a che non venga trovata una soluzione. Nessuno sa se esista un modo migliore per una mappa generica, anche se tante persone hanno provato a trovarne uno.

Posto che Bruno usi una mappa sufficientemente complicata con, per esempio, cinquanta o cento incroci, sembra che nessuno possa decodificare il codice. Anche i matematici più esperti hanno tentato e hanno fallito (c'è però una precisazione da fare: vedi oltre nel paragrafo "Cosa c'entra tutto questo?")

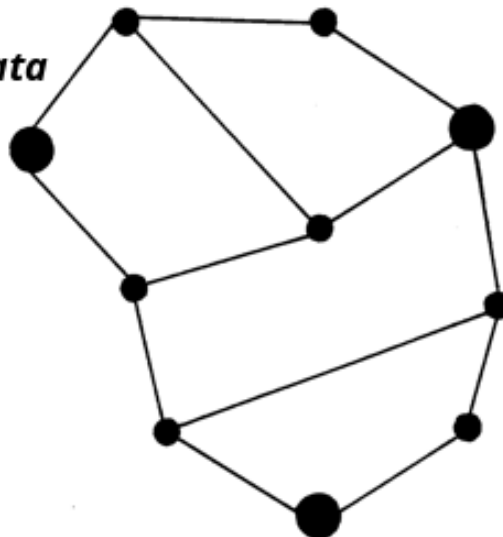
7. Ogni coppia dovrebbe scegliere un "messaggio" (un numero intero), codificarlo con la chiave pubblica e dare la mappa risultante all'altro gruppo. L'altro gruppo dovrebbe tentare di decodificare il messaggio, ma non hanno speranza di poter aver successo se non avrete dato loro la mappa segreta. Quindi date loro la mappa segreta e vedete se ora riescono a decodificare correttamente il messaggio.
8. Ora ogni coppia può disegnare la sua mappa, tenendone riservata la versione segreta e dando la versione pubblica all'altra coppia del gruppo o "pubblicandola" sulla lavagna. Il metodo per costruire le mappe è lo stesso usato nell'attività della città turistica. Potete inserire tutte le strade che volete per nascondere la soluzione. State solo attenti a non inserire strade che iniziano o finiscono in uno dei punti "speciali", in questo modo creereste nuovi incroci che porterebbero a raggiungere il furgoncino dei gelati percorrendo una sola strada. Questo nella situazione della città turistica non sarebbe un problema ma *scompioglierebbe* i calcoli della crittografia. Infatti, i punti speciali non partizionerebbero la mappa in parti che non si sovrappongono, che è il punto essenziale che consente al metodo di funzionare.

Foglio di lavoro: le mappe di Kid Krypto

Mappa Pubblica



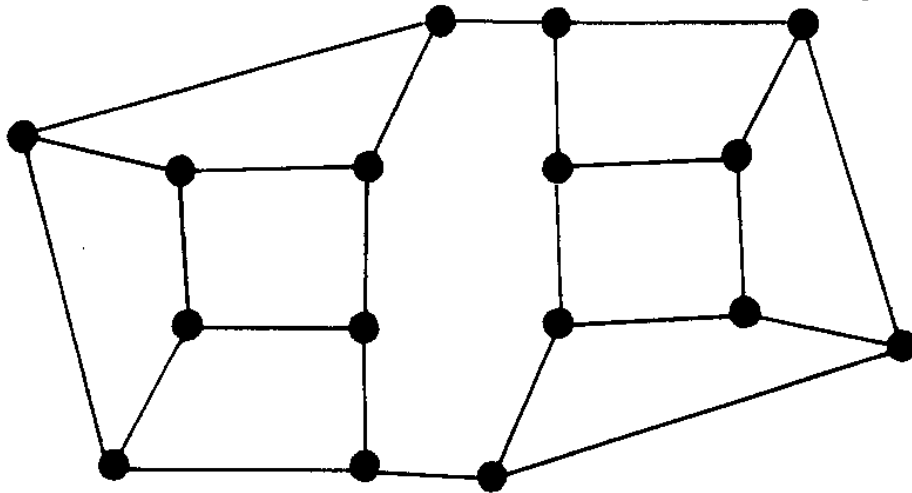
Mappa Privata



Usate queste mappe per criptare e decriptare i messaggi, come indicato nel testo.

Foglio di lavoro: la codifica Kid Krypto

Mostrate questa “mappa” alla classe e usatela per dimostrare la codifica di un messaggio



Cosa c'entra tutto questo?

È chiaro che tutti noi vogliamo spedire messaggi segreti attraverso le reti di computer e nessuno, se non i legittimi destinatari dei nostri messaggi, deve essere in grado di decodificare i messaggi, non importa quanto gli intercettatori siano furbi o quanto tenacemente ci abbiano provato. Naturalmente esistono molti modi per scambiare messaggi riservati quando il mittente e destinatario condividono un codice segreto. Ma la parte intelligente della crittografia a chiave pubblica è che Anna può mandare a Bruno un messaggio in modo riservato senza aver fatto alcun accordo segreto prima, basta che Anna abbia preso il "lucchetto" di Bruno da un luogo pubblico come per esempio da una pagina web.

La riservatezza è solo un aspetto della crittografia. L'altra faccia della medaglia è l'autenticazione. Quando Anna riceve un messaggio da Bruno, come può essere sicura che quel messaggio provenga veramente da Bruno e non da qualche impostore? Supponete che Anna riceva un messaggio di posta elettronica che dice: *"cara Anna, sono bloccato senza soldi, per cortesia accredita 100 euro sul mio conto corrente numero 0241-45-783239. Grazie, ciao. Bruno"*. Alcuni sistemi di crittografia a chiave pubblica possono essere usati anche per questo. In questi sistemi è possibile usare le chiavi anche in senso inverso: usare la chiave privata per criptare e quella pubblica per decriptare. Quindi, come Anna può mandare a Bruno un messaggio segreto criptato usando la chiave pubblica di Bruno, Bruno può criptare un messaggio con la sua chiave privata e spedirlo ad Anna. Se Anna riuscirà a leggerlo usando la chiave pubblica di Bruno, sarà certa che provenga veramente da Bruno. Ovviamente il messaggio di Bruno potrà essere decriptato da tutti quelli che hanno la sua chiave pubblica e quindi non sarà un messaggio riservato, ma sarà autentificato. Se il messaggio da spedire è riservato nulla vieta di criptarlo nuovamente usando la chiave pubblica di Anna. Con questo doppio livello di criptazione si avrà al tempo stesso la certezza che il messaggio provenga veramente da Bruno e che nessuno se non Anna possa leggerlo. Segretezza ed autenticazione vengono forniti dallo stesso metodo a chiave pubblica e privata.

È arrivato il momento di ammettere che mentre lo schema illustrato in questa attività è molto simile a quello usato nei veri metodi crittografici a chiave pubblica, non è in realtà sicuro anche usando mappe molto grandi.

La ragione è che, anche se nessuno conosce il modo ottimale di posizionare i camioncini dei gelati su una mappa arbitraria e da questo punto di vista lo schema è sicuro, accade che ci sia un modo differente per attaccare questo schema. È improbabile che l'idea venga ai vostri studenti, almeno fino alla scuola secondaria, ma è bene che sappiate

che esiste. Si può dire che il metodo illustrato è a prova di studente ma non è a prova di matematico. Per cortesia evitate di leggere il prossimo paragrafo se non siete particolarmente inclini alla matematica.

Numerate gli incroci della mappa $1, 2, 3, \dots, n$. Denotate i numeri assegnati all'inizio da Anna con le lettere $b_1, b_2, b_3, \dots, b_n$ (quelli che sommati assieme danno il messaggio segreto) e indicate con $t_1, t_2, t_3, \dots, t_n$ i numeri trasmessi (quelli indicati fra parentesi). Ora se per esempio l'incrocio 1 è connesso con gli incroci 2, 3 e 4, il numero trasmesso relativo al nodo 1 sarà:

$$t_1 = b_1 + b_2 + b_3 + b_4$$

Naturalmente simili equazioni esistono per ogni altro incrocio. È quindi possibile scrivere le equazioni per ogni nodo ottenendo così un sistema lineare di n equazioni in n incognite che può essere risolto con un programma specifico. In questo modo dai valori $t_1, t_2, t_3, \dots, t_n$ e dalla struttura del grafo si può risalire ai valori delle incognite $b_1, b_2, b_3, \dots, b_n$ e sommando insieme questi ultimi al messaggio originale. Lo sforzo computazionale richiesto per risolvere queste equazioni usando il metodo di eliminazione di Gauss è proporzionale a n^3 , ma queste equazioni sono sparse, cioè molti coefficienti sono nulli e quindi esistono tecniche anche più efficienti. Questo contrasta con lo sforzo computazionale esponenziale necessario, allo stato attuale delle conoscenze, per decrittare la mappa.

Speriamo che non vi sentiate imbrogliati da questa scoperta. Il processi che hanno luogo nei veri sistemi crittografici sono virtualmente identici a quelli qui descritti, ma le tecniche usate per la codifica sono differenti: non sarebbe stato umanamente possibile fare i calcoli a mano degli algoritmi realmente in uso. Il vero metodo di crittografia a chiave pubblica, che è ancora oggi uno di quelli considerati più sicuri, è basato sulla difficoltà di fattorizzare numeri molto grandi.

Quali sono i fattori del seguente numero (composto da 100 cifre)
9.412.343.607.359.262.946.971.172.136.294.514.357.528.981.378.983.082.
541.347.532.211.942.640.121.301.590.698.6 34.089.611.468.911.681 ? È inutile che impieghiate troppo del vostro tempo a fare tentativi, non li troverete.

Sono 86.759.222.313.428.390.812.218.077.095.850.708.048.977 e
108.488.104.853.637.470.612.961.399.842.972.948.409.834.611.525.790.
577.216.753. Non ci sono altri fattori, perché questi due numeri sono primi. Trovare questi fattori dato il loro prodotto è un compito particolarmente arduo: un progetto che impiegherebbe mesi di calcolo per un supercomputer.

In un sistema reale a chiave pubblica Bruno potrebbe usare il numero a 100 cifre come chiave pubblica e i due fattori come chiave privata. La generazione delle chiavi non sarebbe troppo complessa: tutto ciò che

serve è un modo di calcolare numeri primi. Una volta trovati due numeri primi sufficientemente grandi (che non è un problema troppo difficile), si moltiplicano assieme e, oplà, ecco la chiave pubblica. Moltiplicare numeri enormi è un gioco da ragazzi per un computer. Data la chiave pubblica, nessuno potrà trovare quella privata. E se siete preoccupati che qualcuno ci possa riuscire, usate numeri da 200 cifre al posto di quelli da 100 cifre e questo rallenterà di anni il tempo necessario per l'attacco. Nella pratica molto spesso si usano chiavi a 512 bit, che sono grosso modo equivalenti a 155 cifre decimali.

Non abbiamo ancora spiegato come usare i numeri primi per criptare i messaggi usando la chiave pubblica in modo che non si possano decrittare senza la chiave privata. Per fare questo, le cose non sono così semplici come raccontato sopra. Non sono i due numeri primi ad essere utilizzati ma numeri derivati da questi. L'effetto è lo stesso: per violare la codifica occorrerebbe fattorizzare questi numeri. Sarebbe stato possibile tentare di superare le difficoltà e proporre un'attività sulla crittografia a chiave pubblica basata sui numeri primi, ma riteniamo che questo capitolo sia già sufficientemente difficile!

Ma quanto è sicuro un meccanismo basato su numeri primi? La fattorizzazione di numeri molto grandi è stato un problema che ha catturato l'attenzione dei più grandi matematici del mondo per molti secoli e anche se sono stati trovati alcuni metodi per velocizzare la ricerca in modo significativo rispetto alla mera scansione esaustiva delle possibili soluzioni (metodo di forza bruta), nessuno ha trovato un algoritmo realmente veloce, che cioè possa risolvere il problema in tempo polinomiale. Ma dobbiamo fare attenzione. Come abbiamo visto che è possibile violare il codice usato da Bruno senza risolvere il problema della città turistica, ci potrebbero essere modi per violare la codifica basata sui numeri primi senza usare la fattorizzazione. In molti hanno controllato questo metodo e per ora sembra non avere falle.

Un'altra preoccupazione è che ci siano pochi messaggi possibili. Un malintenzionato potrebbe provare a criptarli tutti usando la chiave pubblica del destinatario e confrontare i messaggi intercettati con tutti i messaggi precedentemente criptati. Il metodo usato da Anna evita questo tipo di attacchi perché esistono molti modi diversi di criptare lo stesso messaggio, a seconda di come sono stati scelti i valori che sommati assieme devono dare il valore del messaggio. In pratica i sistemi di crittografia sono sempre progettati in modo che esistano tanti messaggi diversi, anche versioni criptate diverse per lo stesso messaggio, in modo tale che non valga la pena provare a criptarli tutti (e in tutte le varianti di codifica) anche se si disponesse di potentissimi computer.

Non si sa se un metodo efficiente per risolvere la fattorizzazione esista. Nessuno per ora è stato in grado di trovarlo, ma al tempo stesso nessuno è stato in grado di dimostrare che tale metodo non possa

esistere. Se mai un algoritmo efficiente per risolvere il problema della fattorizzazione venisse scoperto, molti dei metodi oggi usati per la crittografia diventerebbero insicuri. Nella parte IV abbiamo discusso i problemi *NP-completi*, che resistono e crollano tutti insieme: se uno di essi diventasse risolvibile in modo efficiente allora tutti lo diventerebbero. Dato che molti sforzi sono stati profusi senza successo nella ricerca di questi algoritmi veloci, essi sembrano essere eccellenti candidati per essere alla base di sistemi crittografici sicuri. Ma è difficoltoso usare uno degli algoritmi *NP-completi* e pertanto i progettisti dei sistemi crittografici continuano a basare i loro metodi su problemi, come la fattorizzazione di un numero, che potrebbero essere più semplici da risolvere degli *NP-completi*, forse molto più semplici. La risposta alla domanda che viene generata da tutto questo vale milioni di euro per il mercato ed è considerata cruciale per la sicurezza nazionale. La crittografia è un'area di ricerca molto attiva nell'ambito dell'informatica.

Per ulteriori approfondimenti

Il libro di Harel, *Algoritmi, lo spirito dell'Informatica* [5], discute la crittografia a chiave pubblica; spiega come usare numeri primi molto grandi per creare un sistema sicuro a chiave pubblica. Un testo molto usato nei corsi di crittografia è *Cryptography and data security* di Dorothy Denning, mentre un altro libro che tratta l'argomento in modo più pratico è *Applied cryptography* di Bruce Schneier [12]. *Turing Omnibus* di Dewdney [3] descrive un altro sistema per realizzare la crittografia a chiave pubblica.

Parte VI

Il volto umano dell'elaborazione — *Interagire con i computer*

Il volto umano dell'elaborazione

Molte persone hanno storie da raccontare in merito a come i computer siano difficili da usare. Sembra che non vogliano mai obbedire e fare ciò che vogliamo noi e che continuino a fare le cose sbagliate e a compiere ridicoli errori. Sembra quasi che i computer siano stati costruiti per dei maghi, non per la gente comune. Ma al contrario dovrebbero proprio essere costruiti in modo che tutti possano utilizzarli, perché sono ormai strumenti fondamentali che ci aiutano ad apprendere, a lavorare e a giocare meglio.

La parte del sistema di elaborazione con la quale interagiamo si chiama "interfaccia utente". È l'elemento più importante! Anche se siamo portati a credere che ciò che il programma fa sia la parte più importante, che l'interfaccia sia solo il modo di comunicare, in realtà un programma diventa inutile se non si riesce ad interagire con esso e fargli fare ciò di cui si ha bisogno. Le interfacce utente sono molto difficili da disegnare ed è stato stimato che nella scrittura dei programmi la parte maggiore degli sforzi sia proprio dedicata all'interfaccia rispetto a tutte le altre parti. Alcuni programmi hanno interfacce grafiche eccellenti, interfacce che non hanno necessità di istruzioni complicate e che diventano praticamente invisibili, vi sentite come trascinati dentro l'applicazione, senza alcuna fatica. Ma innumerevoli programmi che avrebbero potuto avere grande utilità sono stati veri e propri fallimenti perché le loro interfacce erano strane, innaturali e difficili. Intere aziende sono state create su una idea chiara di interfaccia, come il word processor o lo smartphone, che facilita l'accesso alle funzioni di elaborazione.

Ma perché abbiamo proprio bisogno di una interfaccia? Perché non possiamo semplicemente parlare al computer così come parliamo ai nostri amici? Ottima domanda. Forse un giorno potremo, ma certamente non ancora: ci sono grandi limitazioni pratiche su quanto "intelligenti" possano essere i computer di oggi. Le attività che seguono vi aiuteranno a capire i problemi correlati al disegno delle interfacce utente, ad avere una visione più chiara dei limiti dei computer di oggi e a diffidare dei messaggi pubblicitari spesso fuorvianti usati per promuovere strumenti informatici.

Per gli insegnanti

L'informatica non è più tanto legata al calcolo quanto alla comunicazione. L'elaborazione da sola non ha un valore intrinseco, lo assume solo se i risultati possono venire comunicati al mondo esterno e solo in questo caso può avere effetti su di esso. Forse sorprendentemente, questo significa che l'informatica è meno correlata alle macchine che alle persone. Un computer non serve a nulla se non aiuta le persone in qualche modo. Tutte le idee che abbiamo visto su come fare in modo che i computer risolvano problemi in modo

efficiente e rapido sono sensate solo perché le persone hanno necessità che i computer rispondano rapidamente e che sia economicamente vantaggioso usarli.

L'interfaccia è il modo di comunicare fra l'uomo e la macchina. Molte delle attività di questo libro hanno trattato aspetti di comunicazione. La rappresentazione dei dati (parte I) mostra come diversi tipi di informazioni possano essere comunicate ad un computer o fra computer. La rappresentazione delle procedure (parte III) tratta di come comunicare ad un computer le procedure, come cioè dirgli in che modo compiere certi compiti. Dopo tutto, la "programmazione" altro non è se non una spiegazione data ad un computer, usando il suo proprio linguaggio. La crittografia (parte V) descrive come comunicare in segreto o come comunicare elementi segreti senza rivelarli affatto.

Le attività che seguono trattano il problema di come le persone possano comunicare con i computer. Mentre il resto del libro è basato su idee tecniche ben comprese, questa parte è differente. Questo la rende da un lato più facile, perché non sono richieste conoscenze specifiche da parte degli studenti, dall'altro lato è più difficoltosa. Infatti viene richiesta una maggiore maturità per comprendere il senso delle attività proposte ed inserirle in un contesto più ampio. Le spiegazioni fornite a corredo di queste attività sono più lunghe e dettagliate di quelle generalmente presenti altrove nel libro perché è necessario fornire a voi insegnanti sufficiente materiale che vi consenta di essere in grado di guidare le discussioni in classe, riuscendo a gestire alcune delle possibili osservazioni.

Ci sono due attività in questa sezione. La prima è relativa all'area nota come "interfaccia uomo-macchina" (*human-computer interface*), spesso abbreviata con l'acronimo HCI. Per scollegare (*to unplug* come recita il titolo del libro) questi aspetti dell'informatica dalla conoscenza di uno specifico sistema, abbiamo inventato un esercizio di progettazione che non abbia a che fare con i computer ma che nel contempo introduca i principi fondamentali da utilizzare nella progettazione dell'interfaccia uomo-macchina. Siccome la progettazione dell'interfaccia utente è correlata ad aspetti culturali, non esiste alcuna risposta "esatta" per questa attività e questo può essere frustrante per alcuni studenti. La seconda attività è relativa a un tema chiamato "intelligenza artificiale" o AI (*artificial intelligence*). Prevede un gioco che stimola gli studenti a ragionare su cosa un computer possa fare e cosa non possa fare.

Per le menti più tecno-curiose

L'interazione uomo-macchina è diventata una delle aree di ricerca più calde dell'informatica perché è stato compreso che larga parte del successo di un programma dipende dall'interfaccia utente. Questa materia si basa su un'ampia gamma di discipline diverse dall'informatica quali la psicologia, le scienze cognitive, la sociologia e anche

l'antropologia. Pochi studiosi di informatica hanno basi in queste aree e l'interazione uomo-macchina rappresenta un'importante area di crescita per le persone che sono interessate ad aspetti meno tecnologici dell'informatica.

L'intelligenza artificiale è un argomento che spesso genera conflitti e dispute. In questo libro abbiamo tentato di mantenere un equilibrio fra la visione dei sostenitori dell'AI, che credono che le macchine intelligenti siano dietro l'angolo, e gli scettici che sostengono che le macchine non potranno mai essere intelligenti. Il nostro scopo è di incoraggiare gli studenti a sviluppare le loro idee e convinzioni su questi temi e di promuovere una visione equilibrata.

Le attività qui proposte sono basate sulle due testi fondamentali: *La caffettiera del masochista. Il design degli oggetti quotidiani* di Don Norman [10] e *Artificial intelligence: the very idea* di John Haugeland [6]. Vi raccomandiamo questi testi se siete interessati ad approfondire questi temi.

I computer richiedono un altro tipo di comunicazione, che non verrà trattato in questo libro: la comunicazione fra gli informatici. Gli studenti che apprendono tutti i segreti dell'informatica e entrano nel mondo del lavoro, magari dopo aver conseguito una laurea specifica all'università, sono inevitabilmente sorpresi di quanta comunicazione interpersonale sia necessaria per compiere il loro mestiere. I programmi per computer sono gli oggetti più complessi mai costruiti dalla mente umana, con milioni, anzi miliardi di componenti logici che interagiscono fra loro e i progetti di sviluppo del software sono portati avanti da gruppi molto uniti che impiegano molto del loro tempo a comunicare. Una volta che il programma è completato occorre comunicare con gli utilizzatori tramite manuali, corsi, servizi di aiuto telefonico o on-line, senza dimenticare la necessità di comunicare con i potenziali utilizzatori per mezzo di dimostrazioni e materiale pubblicitario. Non abbiamo ancora trovato un modo di descrivere questo aspetto dell'informatica in modo "unplugged" quindi questo problema non verrà trattato nel libro. Questo tema può essere comunque svolto in classe invitando un professionista dell'informatica a descrivere la sua esperienza attivando così la discussione.

Attività 20

La fabbrica di cioccolato — *Progettare l'interfaccia utente*

Sommario

Lo scopo di questa attività è di incrementare la consapevolezza sui problemi correlati alla progettazione di interfacce utente. Poiché che viviamo in un mondo dove la cattiva progettazione è la regola, siamo ormai abituati (rassegnati?) a tollerare i problemi causati dagli strumenti coi quali interagiamo, dando la colpa a noi stessi ("errore umano", "insufficiente addestramento", "è troppo complicato per me") invece che attribuire la colpa a una cattiva progettazione. Questo problema diventa spesso ancora più grave quando si interagisce con i computer. Molti computer infatti vengono progettati per poter svolgere tanti compiti diversi, sono per un uso generale (*general purpose*) e il loro aspetto così la loro interfaccia non fornisce indicazione alcuna su cosa possano fare e su come utilizzarli al meglio.

Abilità

- ✓ Progettazione.
- ✓ Ragionamento.
- ✓ Consapevolezza nell'utilizzo e organizzazione degli oggetti di uso quotidiano.

Età

- ✓ A partire da 7 anni.

Materiale

Ogni gruppo di studenti deve avere:

- ✓ una copia dei fogli di lavoro: "Come si apre quella porta?" (pag. 255) e "il piano cottura" (pag. 256);
- ✓ una copia delle immagini del foglio di lavoro sulle icone, proiettata in aula o fotocopiata su cartoncini che possano essere mostrati alla classe (pag. 257);
- ✓ una o più copie dei sei cartoncini riprodotti nella pagina delle icone. Ritagliate i vari riquadri dal foglio e divideteli fra i gruppi (pag. 258).

La fabbrica di cioccolato

Introduzione

Gli operai della grande fabbrica di cioccolato fanno parte di un popolo di esseri simili ad elfi chiamati Umpa-Lumpa³. Questi Umpa-Lumpa non usano una lingua scritta e hanno una terribile memoria. Per questo hanno difficoltà a ricordare le operazioni che devono compiere per mandare avanti la fabbrica e spesso le cose vanno per il verso sbagliato. Per risolvere questo problema è in corso di progettazione una nuova fabbrica che sarà molto semplice da gestire.

Discussione

1. Spiegate la storia agli studenti e divideteli in piccoli gruppi.
2. Il primo problema che gli Umpa-Lumpa devono affrontare è di entrare e uscire dalle porte portando secchi fumanti di cioccolato liquido. Non riescono a ricordare se le porte sono da spingere, da tirare o se sono scorrevoli. Di conseguenza finisce sempre che si scontrano e che spargono cioccolato appiccicoso tutto attorno. Gli studenti devono compilare il foglio di lavoro intitolato "Come si apre quella porta" a pag. 255. Più di una risposta esatta è presente per ogni porta. Alcune porte (inclusa la prima) non è ovvio come vadano aperte, in questo caso gli studenti dovrebbero annotare che occorre provare. Quando gli studenti hanno completato il proprio modulo, fate in modo che l'intero gruppo discuta il caso di ogni porta, soffermandosi particolarmente su quanto sia facile dire come vada aperta e su come sia facile da usare da chi sta trasportando un secchio di cioccolato. Quindi gli studenti devono decidere quale tipo di porta e di maniglia installare nella nuova fabbrica.
3. Fate seguire a questa attività una discussione in classe. La tabella riportata qui sotto commenta brevemente ogni tipo di porta. Il telaio, i cardini, le maniglie possono fornire indizi su come si apra la porta. Ci sono anche convenzioni per indicare se una porta si apre verso l'interno o verso l'esterno. Identificate il tipo di maniglie in uso nella vostra scuola e discutete se siano o meno appropriate (capita che non lo siano!). Potete portare esempi di porte che spesso vi confondono? Perché? Le porte delle aule si aprono verso l'interno o verso il corridoio? Perché? (Risposta: Spesso si aprono verso l'interno così che la porta non possa colpire le persone che stanno passando nel corridoio, anche se ormai molte porte si aprono verso l'esterno per facilitare la fuga in caso di emergenza).

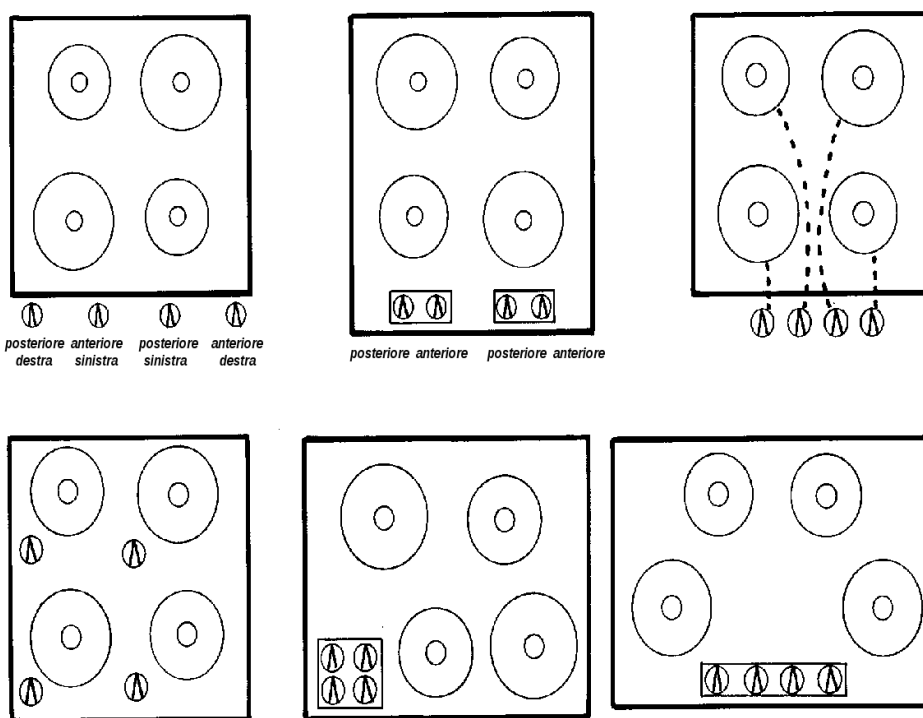
³Ci scusiamo con Roald Dahl. Sicuramente conoscerete gli Umpa-Lumpa se avete letto il libro "Willy Wonka e la fabbrica di cioccolato" o visto il film. Se non l'avete fatto, non preoccupatevi, la trama non è rilevante per lo svolgimento di questa attività.

4. Definiamo un concetto chiave: le *affordance* di un oggetto sono le caratteristiche visibili di un oggetto, sia reali sia percepite, che con la loro parvenza indicano come vada utilizzato. Le *affordance* sono i tipi di una operazione che un oggetto permette (N.d.T. *afford* in inglese). Per esempio, è (normalmente) chiaro dalla sua forma che una sedia serva per sedersi, i tavoli servono per appoggiarvi cose, i pomelli sono da girare, le fessure per inserire qualcosa, i bottoni per premere. Nell'interfaccia usata da un computer, le *affordance* hanno le forme di bottoni, box di testo, menu e così via e danno all'utente indizi su come possano essere usati questi elementi. Se un bottone fosse disegnato come un altro oggetto, le persone non capirebbero che può essere premuto. Questo può sembrare ovvio, ma simili problemi non sono così rari nei dispositivi digitali.

Porta semplice	Non è proprio chiaro come si apra, eccetto per il fatto che, non avendo alcuna maniglia, deve aprirsi spingendo: sarebbe impossibile tirarla.	Porta con cartello	Il cartello è come un piccolo manuale utente. Ma perché una porta dovrebbe essere corredata da un manuale? E poi gli Umpa-Lumpa non sanno leggere
Porta con cardini	Almeno è chiaro quale sia il lato di apertura della porta	Porta con barra	Sembra chiaro che per aprire la porta occorre spingere la barra, ma da quale lato? O forse è da tirare?
Porta con maniglia	Le maniglie di solito indicano che la porta è da tirare, ma potrebbe anche essere una porta scorrevole.	Porta con pomello	Il pomello è una cosa da afferrare, ma non è chiaro se poi occorra spingere o tirare per aprire. È improbabile che questa porta sia scorrevole.
Porta con pannello	È chiaramente una porta da spingere, cosa altro si potrebbe fare?	Porta a vetri	La maniglia verticale è il segnale che la porta va tirata da questo lato, la lunga barra orizzontale dall'altro lato indica che è da spingere.
Porta Scorrevole	Questa non può fare altro che scorrere lateralmente.		

Le porte sono oggetti molto semplici e comuni. Oggetti complessi possono aver necessità di spiegazioni ma non dovrebbe essere il caso per oggetti semplici. Quando oggetti semplici hanno necessità di figure, cartelli o istruzioni vuole dire che sono stati progettati male.

5. I tegami con tipi diversi di cioccolato devono essere cotti a temperature differenti. Nella parte alta del foglio di lavoro “piano di cottura” (pag. 256) vengono mostrati i fornelli e i comandi della vecchia fabbrica di cioccolato. Il pomello a sinistra controlla il fornello anteriore a sinistra, il successivo quello anteriore a destra e i pomelli sul lato destro controllano i fornelli posteriori. Gli Umpa-Lumpa commettono sempre errori e cuociono il cioccolato alla temperatura sbagliata. Inoltre spesso bruciano le loro maniche per andare a regolare i pomelli.
6. Gli studenti dovrebbero ripensare a come sono collocati i pomelli del piano cottura nelle loro cucine di casa e arrivare a proporre una migliore collocazione dei fornelli e dei pomelli per la nuova fabbrica. Fate seguire a questa attività una discussione in classe.



La figura riportata sopra mostra alcune configurazioni comuni di fornelli e pomelli. Nella configurazione in alto a sinistra nella figura ci sono molti modi per far corrispondere i pomelli ai fornelli (precisamente 24 possibilità) e sono necessari cartellini con almeno otto parole (le combinazioni di anteriore, posteriore, destra e sinistra) per indicare la corrispondenza. La disposizione “a coppie” riportata nel caso al centro in alto nella figura è meglio perché ci sono solo quattro possibili corrispondenze. La soluzione in alto a destra specifica la corrispondenza in modo grafico e non linguistico (che è meglio per gli Umpa-Lumpa). I tre disegni in basso non hanno necessità di cartellini o etichette esplicative. La soluzione a sinistra

che prevede un pomello vicino ad ogni fornello è chiaramente scomoda e pericolosa. Le altre due richiedono che vengano spostate le posizioni dei fornelli: nella soluzione proposta al centro lo spostamento si rende necessario per far posto ai pomelli mentre nella soluzione a destra lo spostamento serve per rendere più chiara la corrispondenza.

Il concetto chiave qui è la corrispondenza fra il dispositivo di controllo e l'effetto nel mondo reale. Le corrispondenze naturali, che fanno riferimento ad analogie fisiche o a standard culturali, consentono una comprensione immediata. Come esempio possiamo riportare la corrispondenza spaziale delle tre soluzioni proposte nella seconda riga dell'esempio del piano cottura. Al contrario, corrispondenze arbitrarie, come quelle riportate nei primi tre casi, richiedono cartellini o devono essere spiegate e ricordate.

7. La fabbrica è piena di nastri trasportatori che smistano tegami pieni di cioccolata a vari stadi del processo produttivo. Questi nastri trasportatori sono controllati manualmente dagli Umpa-Lumpa che ricevono ordini da un centro di controllo centrale. Le persone che lavorano al centro di controllo devono poter dire agli Umpa-Lumpa di fermare il nastro trasportatore, di rallentarlo o di riattivarlo.

Nella vecchia fabbrica i comandi venivano dati a voce: un altoparlante vicino alle leve dei nastri trasportatori amplificava la voce degli operatori al centro di controllo. Ma la fabbrica era rumorosa e si sentiva male. Il gruppo dovrebbe disegnare uno schema per fornire i comandi tramite segnali luminosi.

È possibile per esempio fare in modo che i segnali luminosi illuminino scritte come Spegni, Rallenta, Accendi. Gli studenti probabilmente scopriranno che è possibile utilizzare la convenzione normalmente usata dai semafori nelle strade: rosso significa "fermati", giallo "rallenta" e verde "vai". Anche la posizione con il rosso in alto, il giallo nel mezzo e il verde in basso fa parte della convenzione.

Ora rivelate che nella terra degli Umpa-Lumpa i semafori funzionano in modo differente: il colore giallo significa "fermati" e il rosso "vai", il verde appare per indicare che presto dovranno fermarsi. Cosa comporta questo? (Risposta: che nella fabbrica occorrerà seguire la convenzione degli Umpa-Lumpa, non dobbiamo tentare di imporre la nostra).

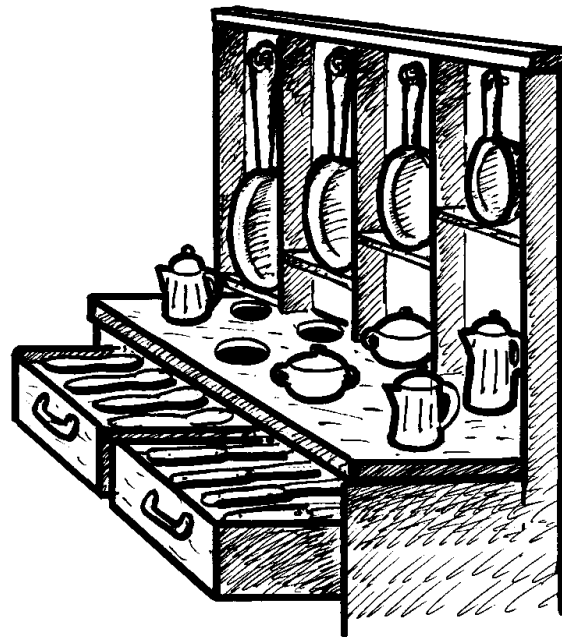
I concetti chiave qui sono: gli *effetti di trasferimento* (le persone trasferiscono le proprie conoscenze e abitudini già acquisite e le applicano in situazioni simili) e gli *stereotipi locali* (diversi popoli apprendono specifici comportamenti e si aspettano che le cose funzionino coerentemente). Anche se l'esempio del semaforo può sembrare strano (nella terra degli Umpa-Lumpa accadono cose bizzarre), ci sono in realtà tanti altri esempi nel nostro mondo reale.

Negli Stati Uniti gli interruttori accendono la luce quando sono rivolti in alto, in Inghilterra accade l'esatto contrario. La tastiera della calcolatrice e quella dei telefoni hanno i tasti in ordine differente: la calcolatrice ha i numeri 1, 2 e 3 in nella fila di bottoni più in basso mentre nel telefono sono in alto. Le date vengono indicate nel mondo in modo differente (Giorno/Mese/Anno, Mese/Giorno/Anno, etc). In Italia si usa la virgola per separare la parte intera da quella decimale di un numero mentre i punti si possono usare per separare le migliaia, negli Stati Uniti il significato dei due simboli è invertito.

8. Quando termina un turno di lavoro degli Umpa-Lumpa alla fabbrica di cioccolato, essi devono riordinare e mettere le ciotole, i tegami, i mestoli, i cucchiari e le spatole al loro posto, pronti per i lavoratori del prossimo turno. C'è una credenza con scaffali dove possono riporre tutti gli attrezzi di lavoro ma all'inizio di ogni turno ci sono sempre difficoltà nel trovare gli strumenti giusti. Gli Umpa-Lumpa, infatti, hanno una cattiva memoria e non riescono a ricordare regole come "metti sempre i tegami nella mensola centrale" oppure "le ciotole vanno messe a destra".

I gruppi di studenti devono tentare di trovare una soluzione migliore. Il diagramma

a destra mostra una buona organizzazione (che è talvolta usata, per scopi differenti, sulle imbarcazioni per evitare che le cose possano scivolare da un lato e dall'altro). Il concetto chiave è di creare vincoli strutturali per rendere ovvia la collocazione di ogni oggetti: l'ampiezza e la sagoma di ogni nicchia rendono chiaro quale utensile vi vada riposto: il progettista ha reso ben visibili le sagome e ha usato le proprietà fisiche degli oggetti evitando così la necessità di doversi rifare a convenzioni da ricordare.



9. Nella sala di controllo principale della fabbrica di cioccolato ci sono tantissime leve e bottoni che comandano specifiche macchine. C'è la necessità di avere targhette che indichino la funzione di ogni leva e bottone, ma siccome gli Umpa-Lumpa non sanno leggere, le targhette devono recare figure, cioè icone, e non parole. Per far

riflettere gli studenti sull'uso delle icone, il foglio di Lavoro con titolo "Icane" (pag. 257) mostra alcuni esempi. Gli studenti dovrebbero identificare il significato delle icone (per esempio la lettera che sta entrando nella buchetta della posta può indicare la spedizione di un messaggio). Non esistono in questo esercizio risposte "esatte", l'idea è semplicemente di identificare quali possano essere i significati.

10. Ora disegnate le icone per la fabbrica di cioccolato. Le schede sul foglio di lavoro delle icone (pag. 258) indicano alcuni gruppi di funzioni correlate e ogni gruppo di studenti dovrebbe ricevere una o più schede senza che gli altri gruppi sappiano quali sono state assegnate. Un pannello di controllo dovrebbe essere disegnato in modo che contenga una icona per ognuna delle cinque o sei funzioni indicate. Gli studenti dovrebbero poi mostrare le icone ai componenti degli altri gruppi, senza indicare la funzione, per vedere se riescono ad individuarne il significato. Incoraggiate l'uso di immaginazione e di colori per creare icone semplici e chiare.

Foglio di Lavoro: Come si apre quella porta?

Riempite il questionario indicando come pensate che queste porte si possano aprire.

PORTA SEMPLICE



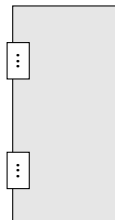
- si apre premendo
- si apre tirando
- è scorrevole
- si apre verso destra
- si apre verso sinistra

PORTA CON CARTELLO



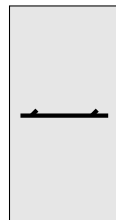
- si apre premendo
- si apre tirando
- è scorrevole
- si apre verso destra
- si apre verso sinistra

PORTA CON CARDINI



- si apre premendo
- si apre tirando
- è scorrevole
- si apre verso destra
- si apre verso sinistra

PORTA CON BARRA



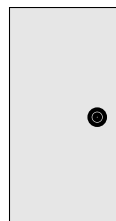
- si apre premendo
- si apre tirando
- è scorrevole
- si apre verso destra
- si apre verso sinistra

PORTA CON MANIGLIA



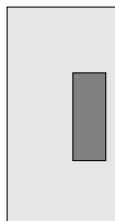
- si apre premendo
- si apre tirando
- è scorrevole
- si apre verso destra
- si apre verso sinistra

PORTA CON POMELLO



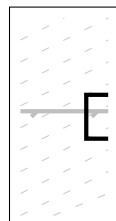
- si apre premendo
- si apre tirando
- è scorrevole
- si apre verso destra
- si apre verso sinistra

PORTA CON PANNELLO



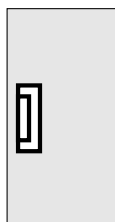
- si apre premendo
- si apre tirando
- è scorrevole
- si apre verso destra
- si apre verso sinistra

PORTA A VETRI



- si apre premendo
- si apre tirando
- è scorrevole
- si apre verso destra
- si apre verso sinistra

PORTA SCORREVOLE

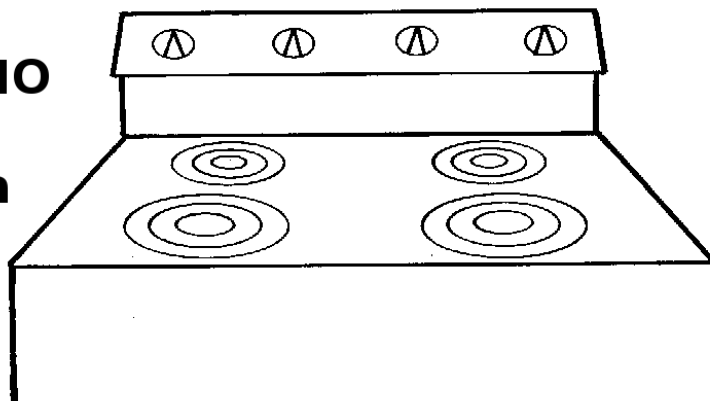


- si apre premendo
- si apre tirando
- è scorrevole
- si apre verso destra
- si apre verso sinistra

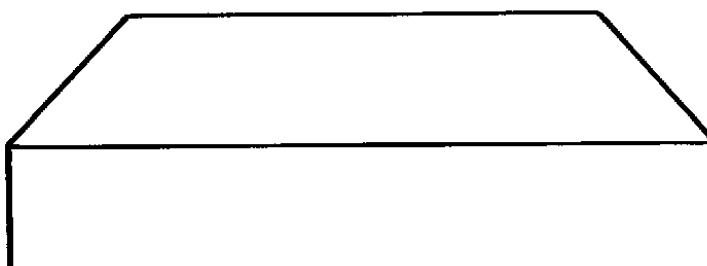
Foglio di Lavoro: il piano cottura

Ridisegnate il piano cottura in modo che sia semplice da usare. Potete aggiungere pannelli frontali o posteriori dove mettere i controlli se lo ritenete necessario per il vostro progetto.

**VECCHIO
piano
cottura**



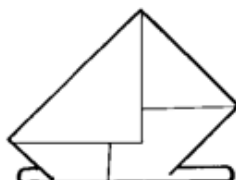
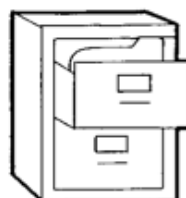
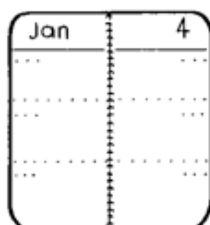
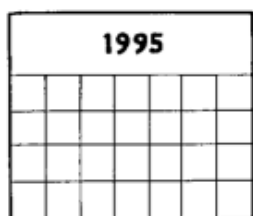
**NUOVO
piano
cottura**



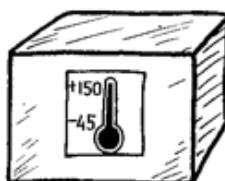
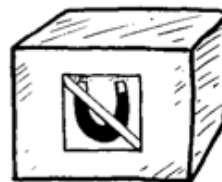
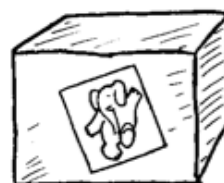
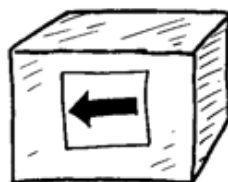
Foglio di Lavoro: Icone

Quale pensate sia il significato di ogni icona/simbolo?

In Ufficio...



Sopra una scatola...



Foglio di Lavoro: le schede delle icone

Ritagliate le schede e datele ad ogni gruppo. Fate in modo che ogni gruppo disegni le icone/i simboli da mettere nel pannello di controllo per rappresentare ogni istruzione.

Ingredienti

- aggiungi • cacao
- latte
- zucchero
- più zucchero
- burro

Extra

- aggiungi • nocciole
- caramello
- zenzero
- uvetta
- cocco

Preparazione

- inizia a miscelare
- finisci di miscelare
- inizia a cuocere
- finisci di cuocere
- versa negli stampi
- disegna una figura (tante figure diverse!)

Assaggio

- assaggia
- ottimo! prima scelta
- buono! seconda scelta
- cioccolato cotto un po' male
- disgustoso, da buttare

Formati

- tavoletta piccola
- tavoletta media
- tavoletta grande
- tavoletta enorme
- indica l'ampiezza (in quadretti)
- fai scaglie di cioccolato

Impacchettamento

- avvolgi nella stagnola
- avvolgi nella carta
- metti in un sacchetto
- metti in una scatola
- attiva il nastro trasportatore
- ferma il nastro trasportatore

Variazioni ed Estensioni

Gli studenti riescono ad impostare l'ora esatta nel proprio orologio da polso o nel forno a microonde? Il progetto del piano di cottura è stato semplice perché ci sono solo quattro fornelli e quattro pomelli. Problemi più complessi emergono quando il numero delle azioni è molto maggiore del numero dei dispositivi di controllo. Dare indicazioni agli orologi da polso elettronici o ai forni a microonde è spesso difficile non per il numero dei bottoni (che sono pochi) ma per il numero degli stati nei quali l'apparato può trovarsi. ("Ci vorrebbe una laurea in ingegneria dell'MIT per usare questo", disse qualcuno a Don Norman, un grande psicologo delle interfacce utente, guardando il proprio nuovo e ipertecnologico orologio da polso. Don ha anche una laurea in ingegneria dell'MIT e, in poche ore, è stato in grado di usare perfettamente l'orologio. Ma perché mai operazioni di questo tipo dovrebbero richiedere ore?

Gli studenti dovrebbero riflettere sui casi che conoscono nei quali hanno visto persone confuse o frustrate per l'uso di apparati elettronici. Telefoni cellulari, videoregistratori, computer, telecomandi, tutti questi apparati hanno la capacità di generare frustrazione. Gli studenti dovrebbero domandarsi quale è il dispositivo che confonde gli utenti e come fare per poterne disegnare meglio l'interfaccia.

Cosa c'entra tutto questo?

L'interazione Uomo-Macchina richiede un'attenta progettazione, valutazione e realizzazione di sistemi che consentano alle persone di svolgere le loro attività in modo sicuro e produttivo. Un tempo i computer erano solo per specialisti e ci si aspettava che venissero usati solo da persone molto istruite e con una specifica preparazione. Questo convincimento poi sparì e divenne completamente normale comperare i libri per principianti (for "dummies") per imparare ad usare i propri computer. Oggi i computer sono strumenti che tutti possono usare e grande attenzione viene posta nella progettazione dell'interfaccia utente.

Molti disastri, inclusi alcuni che hanno comportato la perdita di vite umane, sono stati causati da interfacce inadeguate: incidenti aerei o abbattimenti di velivoli civili, tamponamenti autostradali causati da erronee indicazioni sui pannelli elettronici, disastri nucleari. In una scala più piccola, la maggior parte delle persone si sente frustrata, spesso molto frustrata nell'interazione con i computer e con gli altri strumenti elettronici presenti sul proprio luogo di lavoro (un poliziotto arrivò a sparare al proprio computer). E il problema non è limitato ai dispositivi elettronici. Provate a pensare ai contenitori che non si aprono se non con forbici appuntite o pinze, alle porte che vi fanno male ai polsi tutte le volte che le usate, al contenitore del latte che vi schizza ogni volta che lo aprite, ai sistemi audio-video, stereo, decoder che nelle loro pubblicità dovrebbero avere mille funzioni ma che nella realtà rendono complicatissime anche le operazioni più comuni.

Ci stiamo abituando agli "errori umani". Le persone spesso danno la colpa a se stesse quando le cose non funzionano ma i cosiddetti "errori umani" sono nei fatti errori di progettazione. Le persone hanno dei limiti nella quantità di informazioni che riescono ad elaborare e i progettisti devono tenere questo fatto nella dovuta considerazione. Una cattiva progettazione non si può rimediare fornendo un manuale utente dettagliato e complicato ed aspettarsi che le persone lo studino in tutti i particolari e lo ricordino per sempre. Inoltre gli esseri umani possono sbagliare. Anche di questo fatto occorre tenere conto nella progettazione.

La valutazione delle interfacce è una parte essenziale del processo di progettazione. Il lavoro svolto in questa attività ha coinvolto alcune fasi di valutazione quando gli studenti hanno mostrato le icone disegnate dal proprio gruppo agli altri. Una valutazione più approfondita avrebbe richiesto la sperimentazione con Umpa-Lumpa reali (che potrebbero percepire le icone in modo diverso) attraverso un esperimento controllato, come fanno gli psicologi.

Anche se i problemi causati dalla tecnologia spesso forniscono materia per molte barzellette, la progettazione delle interfacce non è per nulla un argomento sul quale ridere. Interfacce inadeguate possono causare problemi che vanno dall'insoddisfazione nel luogo di lavoro fino alle cadute in borsa, dalla perdita dell'autostima alla perdita della vita.

Per ulteriori approfondimenti

Il libro di Don Norman *La caffettiera del masochista. Il design degli oggetti quotidiani* [10] è una piacevole (e liberatoria) disamina della miriade di problemi di progettazione che sono presenti negli oggetti della nostra vita quotidiana. *Designing with the mind in mind* di Jeff Johnson [7] presenta una visione introspettiva su come le persone pensino e come le interfacce dovrebbero essere progettate per tenere conto dell'elemento umano.

Attività 21

Conversazioni coi computer — *Il test di Turing*

Sommario

Questa attività mira a stimolare la discussione sulla domanda se un computer possa essere “intelligente” o se sia possibile che lo diventi in futuro. Basata sulla visione di un pioniere della ricerca in informatica di come sia possibile individuare l’intelligenza artificiale, se mai comparisse, questa attività mostra come sia attualmente possibile e facile essere ingannati da dimostrazioni di “intelligenza” scelte con cura.

Abilità

- ✓ Saper intervistare.
- ✓ Capacità di ragionamento.

Età

- ✓ A partire da 7 anni.

Materiale

- ✓ Una copia delle domande del Foglio di lavoro del Test di Turing (una copia per ogni coppia di studenti oppure una copia da proiettare);
- ✓ una copia del foglio delle risposte al Test di Turing.

Conversazioni coi computer



Discussione

Questa attività prende la forma di un gioco nel quale gli studenti tentano di distinguere fra un essere umano e un computer facendo domande ed analizzando le risposte.

Ci sono quattro attori che chiameremo Iacopo, Irene, Ugo e Carla (la prima lettera del nome aiuterà a ricordare il ruolo di ogni protagonista). L'insegnante coordina lo svolgimento dell'attività. Il resto della classe partecipa come pubblico attivo della rappresentazione. Iacopo e Irene sono intermediari, Ugo fornirà personalmente le risposte alle domande come essere umano mentre Carla fingerà di essere un computer. La classe dovrà scoprire quale dei due è l'essere umano e quale il computer. Iacopo e Irene avranno il ruolo di garantire che il gioco sia corretto: essi porteranno le domande a Ugo e Carla che saranno in due stanze differenti e separate dall'aula dove è il resto della classe.

Iacopo prende le domande della classe e le porta a Ugo mentre Irene porta le stesse domande a Carla (senza che il resto della classe possa sapere chi stia portando le domande a chi). È necessario avere intermediari perché non si vedaano Ugo e Carla mentre rispondono alle domande.

Prima che la classe inizi questa attività, selezionate le persone che devono ricoprire questi ruoli e brevemente spiegate loro cosa devono fare. Iacopo e Irene devono portare le domande della classe rispettivamente a Ugo e Carla e portare indietro le loro risposte alla classe. È importante che non rivelino chi sia il loro interlocutore, per esempio dicendo "Lui dice che...". Ugo scrive le sue personali risposte alle domande, esprimendosi in modo breve, accurato e onesto. Carla

risponde alle domande cercando le domande sul foglio delle risposte e copiando le risposte lì indicate. Per alcune domande Carla deve costruire la risposta seguendo le istruzioni scritte in corsivo sul foglio delle risposte.

Iacopo e Irene dovrebbero essere forniti di carta e penna perché alcune risposte potrebbero essere difficili da ricordare.

1. Prima di iniziare il gioco raccogliete le opinioni degli studenti sulla questione se i computer siano intelligenti o se gli studenti pensino che un giorno possano diventare tali. Domandate come si possa fare, secondo loro, a determinare se un computer sia intelligente o meno.
2. Fate un'introduzione al test di "intelligenza" nel quale dovranno riconoscere un essere umano da un computer solo facendo domande. Il computer passa il test se la classe non è in grado di capire la differenza in modo affidabile. Spiegate che Iacopo e Irene porteranno le domande a due persone una delle quali fornirà la sua (umana) risposta mentre l'altra darà la risposta che un computer può dare. Lo scopo del gioco è di individuare chi sta dando le risposte da computer.
3. Mostrate agli studenti la lista delle domande possibili dal foglio *Domande del Test di Turing*, distribuendo fotocopie o proiettando l'elenco.

Lasciate che gli studenti scelgano quali domande debbano essere formulate per prime. Una volta scelta la domanda fatevi spiegare perché pensano che quella domanda sia importante per distinguere fra il computer e l'essere umano. Questo ragionamento è la parte più importante dell'esercizio perché obbligherà gli studenti a pensare a quali risposte possa dare un essere umano che al tempo stesso non possano essere fornite da un computer.

La classe dovrebbe discutere quindi su quale risposta sia plausibile che sia stata generata da un computer.

Ripetete il procedimento per alcune domande, preferibilmente fino a quando la classe non sia sicura di aver individuato quale sia il computer. Se la classe trova la soluzione troppo rapidamente, fate ripetere l'esperimento facendo giocare Iacopo e Irene a testa o croce, così che la classe non possa sapere questa volta chi porta i messaggi all'essere umano e chi al computer. Le risposte che Carla fornisce non sono molto diverse da quelle che potrebbe generare un computer "intelligente". Alcune delle domande possono facilmente rivelare il computer. Per esempio è difficile che un essere umano sia in grado di fornire il valore della radice di due fino alla ventesima cifra decimale. La maggior parte delle persone non sono assolutamente in grado di rispondere alla domanda. Alcune domande possono rivelare chi sia il computer quando le risposte

vengono analizzate insieme. Per esempio le risposte alle domande "Ti piace..." possono essere plausibili singolarmente, ma quando se ne analizzano diverse può apparire evidente la semplice formula usata per generare la risposta. Alcune delle risposte indicano che la domanda è stata fraintesa, per indurre la classe a pensare che sia stato l'essere umano a commettere l'errore.

Molte delle risposte sono generiche, ma sicure. Probabilmente una domanda successiva e che prende spunto dalla risposta stessa potrebbe facilmente svelare che il computer non sta realmente capendo il significato. La risposta "non lo so" è ragionevolmente sicura per il computer e lo potrebbe fare apparire anche più umano: potremmo aspettarci che anche gli studenti rispondano "non lo so" ad alcune delle domande, come per esempio quella sul valore della radice di due. Comunque se il computer desse questa risposta troppo spesso, anche a domande semplici, rivelerebbe la sua identità.

Dato che lo scopo del computer è di far pensare a coloro che fanno le domande di star avendo a che fare con un essere umano, alcune delle risposte sono volutamente fuorvianti, come le risposte a problemi aritmetici scorrette o fornite con ritardo. Le domande e le risposte dovrebbero fornire parecchio materiale per la discussione.

Foglio di lavoro: Le domande del Test di Turing

Scegliete da questa lista le domande da formulare all'essere umano e al "computer".

1. Qual è il nome della sorellina piccola di Bart Simpson?
2. Cosa pensi di Roald Dahl?
3. Sei un computer?
4. Data la sequenza 3, 6, 9, 12, 15, qual è il numero successivo?
5. Cosa pensi delle armi nucleari?
6. Quanto fa 2×78 ?
7. Quanto vale la radice quadrata di due?
8. Somma 34957 e 70764.
9. Ti piace la scuola?
10. Ti piace ballare?
11. Che giorno è oggi?
12. Che ore sono?
13. Quanti giorni ha il mese di febbraio negli anni bisestili?
14. Quanti giorni ci sono in una settimana?
15. Quale nazione ha come bandiera un cerchio rosso su sfondo bianco?
16. Ti piace leggere libri?
17. Qual è il tuo cibo preferito?

Foglio di lavoro: Le risposte del Test di Turing

Cerca la domanda su questo foglio e rispondi come indicato.

1. Qual è il nome della sorellina piccola di Bart Simpson?

Non me lo ricordo.

2. Cosa pensi di Roald Dahl?

Che scrive libri divertenti

3. Sei un computer?

E tu sei un computer?

4. Data la sequenza 3, 6, 9, 12, 15, qual è il numero successivo?

18.

5. Cosa pensi delle armi nucleari?

Le armi nucleari sono molto pericolose e non dovrebbero mai essere usate

6. Quanto fa 2×78 ?

166 (la risposta è volutamente errata)

7. Quanto vale la radice quadrata di due?

1.41421356237309504878

8. Somma 34957 e 70764.

**aspetta circa 20 secondi prima di dare la risposta
...105621.**

9. Ti piace la scuola?

Certamente

10. Ti piace ballare?

Sì, mi piace ballare

11. Che giorno è oggi?

Fornisci il giorno della settimana corretto

12. Che ore sono?

Fornisci l'ora esatta

13. Quanti giorni ha il mese di febbraio negli anni bisestili?

il 2012 e il 2014 sono stati anni bisestili. (la risposta è volutamente errata)

14. Quanti giorni ci sono in una settimana?

Sette.

15. Quale nazione ha come bandiera un cerchio rosso su sfondo bianco?

Non lo so

16. Ti piace leggere libri?

Sì, mi piace leggere.

17. Qual è il tuo cibo preferito?

Non ho fame, grazie.

Variazioni ed estensioni

Occorrono almeno tre persone per giocare. Irene può interpretare anche il ruolo di Iacopo e di Carla. Irene porta le domande a Ugo, annota sia le risposte di Ugo sia quelle del foglio di lavoro *le risposte del test di Turing*. Irene restituisce le due risposte usando le lettere A e B per identificare quali risposte sono dello stesso autore.

Per capire se un computer possa emulare un umano rispondendo alle domande, considerate con la classe quali siano le conoscenze necessarie per rispondere ad ogni domanda prevista dal test. Gli studenti potrebbero proporre altre domande che avrebbero voluto chiedere e si può discutere quali avrebbero potuto essere le possibili risposte. Questo richiede immaginazione perché è impossibile prevedere quali sviluppi possa avere la conversazione.

Per illustrare la situazione qui a fianco ci sono due conversazioni di esempio. La prima è basata su domande “fattuali” alle quali un computer può rispondere correttamente, mentre la seconda mostra come una conversazione possa coinvolgere molti argomenti e dimostra l’ampiezza della conoscenza che sarebbe necessaria ad un computer per poter sostenere la conversazione.

C’è un programma chiamato “Eliza” che è facilmente disponibile sul Web e crea una specie di chat testuale attraverso la quale si possono avere conversazioni. Eliza simula la seduta di uno psicoterapeuta e può generare conversazioni che appaiono notevolmente intelligenti seguendo alcune semplici regole. Alcuni esempi di conversazioni con Eliza sono descritte più avanti. Gli studenti possono provare Eliza o altre chatbot (chat automatiche), anche se

Domanda: Per cortesia, scrivi un sonetto sul Forth Bridge.
Risposta: Non è cosa per me. Non sono capace di scrivere poesie.
Domanda: Somma 34957 e 70764.
Risposta: attende 30 secondi e poi... 105621.
Domanda: Sai giocare a scacchi?
Risposta: Sì.
Domanda: Il mio Re è in K1 e non ho altri pezzi. Tu hai solo il Re in K6 e una torre in R1. Qual è la tua mossa?
Risposta: ...dopo una pausa di 15 secondi... Torre in R8, scacco matto.

Domanda: Se nel primo verso del sonetto che dice “Posso paragonarti a un giorno d’estate?” mettessimo “giorno di primavera”, sarebbe lo stesso o forse meglio?
Risposta: Non suona bene.
Domanda: E allora “un giorno d’inverno”? Questo suona bene.
Risposta: Sì, ma nessuno vorrebbe essere paragonato a un giorno d’inverno.
Domanda: Potresti dire al sig. Pickwick, che ti ricorda il Natale?
Risposta: In un certo senso.
Domanda: Ma Natale è un giorno di inverno e io non penso che al Sig. Pickwick dispiacerebbe il paragone.
Risposta: Non penso che tu sia serio. Con giorno d’inverno uno intende un giorno di inverno normale, non uno speciale come Natale.

occorre fare attenzione perché
alcune possono trattare argomenti o usare linguaggi non adatti ai
ragazzi in età scolare.

Cosa c'entra tutto questo?

Pes secoli i filosofi hanno dibattuto il problema se una macchina possa simulare l'intelligenza umana e, dall'altro lato, se la mente umana sia solamente una macchina che esegue uno splendido programma. Questa questione divide nettamente le persone. Alcuni trovano l'idea pretenziosa, una pazzia o anche blasfema, mentre altri pensano che l'intelligenza artificiale sia inevitabile e che alla fine saremo in grado di costruire macchine che saranno intelligenti come noi. Molti film di fantascienza hanno fatto notare che se le macchine poi fossero in grado di superare la nostra intelligenza potrebbero creare macchine ancora più intelligenti. I ricercatori dell'intelligenza artificiale sono stati criticati per aver usato i loro progetti troppo ambiziosi come mezzo per attrarre finanziamenti dai governi i quali vorrebbero creare macchine da guerra autonome. I ricercatori, dal loro canto, etichettano le proteste come reazioni luddite e indicano i chiari effetti positivi per la società portati da ogni piccolo incremento di "intelligenza" degli strumenti attorno a noi. L'intelligenza artificiale, in una visione equilibrata, non è nè pretenziosa nè inevitabile: mentre ora nessun programma può essere considerato "intelligente", nel senso ampio del termine, la risposta alla domanda se un giorno ciò sarà possibile deve essere data in modo sperimentale, non in altro modo.

Il dibattito sulla intelligenza artificiale dipende dalla definizione di intelligenza. Molte definizioni sono state date e discusse. Un modo interessante di stabilire il concetto di intelligenza come "esperimento teorico" è stato proposto nei tardi anni '40 del ventesimo secolo da Alan Turing, un eminente matematico inglese, attivo nel controspionaggio durante la seconda guerra mondiale e anche maratoneta. L'approccio di Turing è operativo: invece di definire l'intelligenza, Turing definisce una situazione per fare in modo che un computer possa dimostrare di essere intelligente. Lo scenario è simile a quello descritto nella precedente attività. Il punto chiave è avere una persona che formula domande interagendo con un essere umano e con una macchina usando una telescrivente (la tecnologia di avanguardia nel 1940). Se la persona che fa le domande non riesce a distinguere quali risposte provengano da un essere umano e quali dalla macchina allora il computer ha superato il test di Turing per l'intelligenza. L'uso della telescrivente evita il problema che il computer venga riconosciuto dalle caratteristiche fisiche o dal tono di voce. Si potrebbe pensare ad estendere questo esercizio così che la macchina debba riuscire ad imitare una persona nell'aspetto, nel suono, al tatto e anche nell'odore, ma questi aspetti fisici appaiono poco rilevanti per il concetto di intelligenza.

Il test di Turing originale era un po' differente dal nostro. Un esercizio preliminare nel test originale prevedeva che venissero fatte domande

ad un uomo e a una donna e l'interrogante doveva stabilire il genere degli interlocutori. L'uomo doveva tentare di convincere chi faceva le domande di essere una donna mentre la donna tentava di convincerlo di essere lei effettivamente la donna. Turing quindi prevede di sostituire uno dei due con un computer (e questo per l'epoca era solo una ipotesi) per vedere se fosse altrettanto convincente nel fingere di essere una persona in questo gioco d'imitazione (*imitation game* come il titolo di un film del 2014). Abbiamo cambiato la struttura per questa attività scolastica perché il tipo di domande che gli studenti avrebbero formulato per individuare il genere avrebbero potuto non essere appropriate e avrebbero potuto promuovere stereotipi sessuali. Imitare l'intelligenza è difficile. Ma anche se i ruoli fossero invertiti e una persona volesse farsi passare per un computer, probabilmente non ci riuscirebbe: verrebbe certamente svelato l'inganno da risposte lente e probabilmente inaccurate a domande come "quanto fa 123456 per 789012?"

E' al contrario sorprendentemente semplice per un computer acquisire una sottile parvenza di capacità di dialogo. Un famoso programma sviluppato agli inizi degli anni 60, Eliza (già citato), simulava uno psicoterapeuta che dialoga con un paziente. Qui a destra viene riportato l'esempio tradotto di una conversazione con il sistema. L'apparenza di una conversazione naturale viene generata con meccanismi quali:

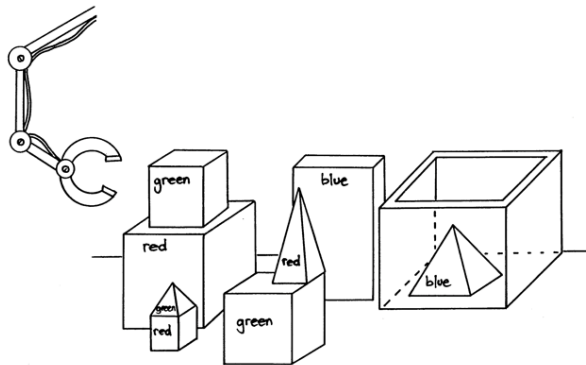
1. usare risposte preconfezionate (per esempio: *Può dirmi un esempio specifico?*, in risposta a frasi come *Sono sempre...; Mi dispiace sapere che... in risposta a Dice che...;*)
2. ripetere la frase del paziente (per esempio: *Il suo fidanzato l'ha obbligata a venire qui* in risposta a *Il mio fidanzato mi ha obbligato a venire qui*)
3. riconoscere parole chiave (*Mi dica di più della sua famiglia... al riconoscimento delle parola madre*);
4. alcune frasi standard *Cosa significa per lei ...;*

Persona:	Gli uomini sono tutti uguali.
Computer:	In che senso?
Persona:	Ci annoiano sempre con una cosa o un'altra.
Computer:	Può dirmi un esempio specifico?
Persona:	Il mio fidanzato mi ha obbligato a venire qui.
Computer:	Il suo fidanzato l'ha obbligata a venire qui.
Persona:	Dice che sono molto spesso depressa.
Computer:	Mi dispiace sapere che è depressa.
Persona:	È vero. Sono infelice.
Computer:	Pensa che venire qui l'aiuterà a non essere invelice?
Persona:	Ho bisogno di aiuto, questo sembra certo.
Computer:	Cosa significa per lei avere aiuto?
Persona:	Forse potrei imparare ad andare d'accordo con mia madre.
Computer:	Mi dica di più della sua famiglia...
Computer:	Cosa altro le viene in mente quando pensa a suo padre?
Persona:	La prepotenza.
Computer:	E questo ha qualcosa a che fare con il fatto che il suo fidanzato l'ha obbligata a venire qui?

5. riprendere argomenti già trattati nella conversazione (*E questo ha qualcosa a che fare con...*);

Naturalmente questo non significa che il test di Turing sia stato superato, perché la persona non sta facendo domande per determinare se l'interlocutore sia una macchina o no; se lo facesse non ci vorrebbe molto per scoprirlo.

Un altro sistema che sembra essere in grado di intrattenere una conversazione intelligente è il programma chiamato 'SHRDLU', sviluppato alla fine degli anni '60. Esso prendeva in input istruzioni per manipolare blocchi posti su di un tavolo.' I blocchi erano simulati e si potevano vedere come immagini sullo schermo di un computer, anche se non sarebbe stato difficile far fare le operazioni ad un robot reale (almeno in via di principio). La figura mostra il micro-mondo del robot.



SHRDLU può sostenere una conversazione sorprendentemente fluente, come si può vedere dal dialogo tradotto nella pagina seguente. Fra le altre cose questo illustra come possa obbedire ad ordini, disambiguare istruzioni, riconoscere a cosa fanno riferimento i pronomi e apprendere nuovi termini ("campanile").

Persona: Prendi il grande blocco rosso.

Robot: OK.
Il sistema risponde "OK" quando svolge un compito. Per prendere il blocco rosso deve togliere prima il blocco verde che è sopra e per fare questo deve trovare un posto dove metterlo.

Persona: Afferra la piramide.

Robot: Non so a quale piramide tu ti riferisca.
Ci sono tre piramidi nella scena, quindi il programma non fa nulla.

Persona: Cerca un blocco più alto di quello che stai tenendo e mettilo dentro la scatola.
Il programma deve dedurre cosa significhi "quello" e a cosa si riferisca la particella "lo" di "mettilo".

Robot: Assumo che ti riferisca al blocco che è più alto di quello che sto tenendo.
La risposta serve per tenerci informati.

Robot: OK.
Il braccio del robot deve appoggiare il blocco rosso e prendere il blocco blu che è più alto.

Persona: Cosa c'è nella scatola?

Robot: La piramide blue e il blocco blu.

Persona: Su cosa appoggia la piramide?
Prima il programma ha risposto ad una richiesta sulla "piramide" chiedendo a quale piramide si riferisse. Ora sa che si fa riferimento alla piramide appena citata.

Robot: Sulla scatola.

Persona: Un campanile...

Robot: Scusa, non capisco la parola "campanile".
Il sistema interrompe immediatamente quando vede una parola che non riesce ad interpretare.

Persona: Un "campanile" è una pila che contiene due cubi verdi e una piramide.
Le virgolette indicano che si sta definendo una nuova parola.

Robot: Capisco.

Persona: Ci sono campanili adesso?

Robot: No.
La nuova definizione viene usata per riconoscere i campanili.

Persona: Costruiscine uno.

Robot: OK.
La descrizione data nella definizione è sufficiente per creare strategie per riconoscere e costruire oggetti del nuovo tipo. La parola "uno" significa "un campanile" e il lavoro viene compiuto.

Il sistema è comunque molto fragile e l'illusione che crea viene facilmente distrutta. Qui a fianco viene riportata una continuazione della conversazione che non è stata generata dal programma, ma che avrebbe potuto esserlo. Anche se è stato possibile definire il nuovo termine "campanile" nella precedente conversazione, solo una ristretta gamma di termini possono essere descritti al robot che ha una "intelligenza" estremamente limitata.

Alla fine deve ammettere che non conosce il senso delle parole che lui stesso usa: (*Scusa, ma non conosco la parola "scusa"*)

Ora considerate una continuazione alternativa, completamente immaginaria, che potrebbe essere stata generata da uno studente. Sebbene minimo, questo frammento dimostra una ricca conoscenza del mondo.

- ✓ Il "robot" sa che una pistola ad acqua ha minor valore se se ne possiede già una.
- ✓ Inoltre, si aspetta che anche la persona lo sappia perché non si preoccupa di spiegare che quello è il motivo per declinare l'offerta.
- ✓ Ancora, comprende che il desiderio della persona per un blocco è sufficiente per rinunciare a qualcosa di importante.
- ✓ In più, riconosce che, al contrario delle pistole, il valore dei blocchi aumenta con la quantità.
- ✓ Tenta di ottenere una situazione favorevole, denigrando la rana.
- ✓ Ciò implica che è disposto ad iniziare una trattativa.

Certamente questo grado di scaltrezza va oltre la capacità dei computer attuali!

Persona:	Permuta la tua pistola ad acqua con il grande blocco rosso.
Robot:	Scusa, non conosco la parola "permuta".
Persona:	Uno "permuta" è un libero scambio di proprietà.
Robot:	Scusa, non conosco la parola "libero".
Persona:	Un atto "libero" viene fatto per buona volontà senza essere obbligati.
Robot:	Scusa, non conosco la parola "atto".
Persona:	Un "atto" è ciò che una persona fa volontariamente, non per caso.
Robot:	Scusa, non conosco la parola "persona".
Persona:	Scusa, ma pensavo fossi più intelligente di quanto tu sia.
Robot:	Scusa, ma non conosco la parola "scusa".

Person:	Permuta la tua pistola ad acqua con il grande blocco rosso.
Robot?	Ho già una pistola ad acqua, ma ti darò due blocchi rossi e una piramide in cambio della tua viscida vecchia rana.

La storia sulla destra è stata generata da un altro programma (della fine degli anni '80). Anche se appare subito chiaramente che è stato generato da un computer per la sua freddezza e concisione, non è difficile pensare a come potrebbe essere completato aggiungendo ogni sorta di dettagli ulteriori. Quello che è

interessante qui non è l'aspetto superficiale della storia ma il racconto in essa contenuto. Anche se questo racconto è molto lontano da quanto possa essere generato da una persona, sembra catturare alcuni elementi umani di conflitto. Oggi ci sono tanti sistemi per la generazione automatica di storie, anche se la sfida è quella di determinare quanto del materiale sia solo la mera giustapposizione di spezzoni di racconto legati assieme e quanto invece sia stato costruito in modo creativo come nell'esempio qui sopra.

C'era una volta una sterna artica chiamata Truman. Truman non aveva una casa. Truman aveva necessità di un nido. Truman volò sulla spiaggia. Truman cercava rametti. Volò nella tundra. Incontrò un orso polare chiamato Horace. Truman domandò a Horace dove poteva trovare dei rametti. Horace nascose i rametti. Horace disse a Truman che c'erano rametti sull'iceberg. Truman volò verso l'iceberg. Cercò i rametti. Non trovò rametti. Horace cercava della carne. Trovò della carne. Mangiò Truman. Truman morì.

C'è una competizione annuale chiamata "premio Loebner", nel quale i programmi gareggiano per passare il test di Turing riuscendo a convincere la giuria di essere persone. Fino al 2014 nessun programma è riuscito a vincere la medaglia d'oro né quella d'argento, ma è stata riconosciuta ogni anno una medaglia di bronzo al programma che viene giudicato il più "umano". Nella competizione del primo anno (1991) ha vinto un programma che, fra gli altri trucchi, metteva errori di digitazione per apparire più umano!

Nessun sistema di intelligenza artificiale è stato creato ad oggi che sia neanche prossimo a passare completamente il test di Turing. Anche se ciò un giorno succedesse, i filosofi hanno osservato che questo test non misurerebbe tuttavia realmente ciò che le persone intendono con "intelligenza". Il test fornisce infatti solo una equivalenza comportamentale: è stato disegnato per determinare se un programma mostra i sintomi dell'intelligenza, che non è la stessa cosa rispetto al reale possesso dell'intelligenza. Ci può essere intelligenza umana senza consapevolezza, conoscenza di sé, auto-coscienza, senza poter provare amore, sentire di essere ... vivi?

Verosimilmente il dibattito sull'intelligenza artificiale ci farà compagnia ancora per molti decenni.

Per ulteriori approfondimenti

Intelligenza Artificiale Il Significato Di Un'Idea del filosofo John Haugeland [6] è un libro divulgativo sul dibattito dell'intelligenza artificiale ed è la fonte di alcune delle illustrazioni di questa attività (in particolare la conversazione di SHRDLU e la relativa discussione).

Turing ha descritto il test che porta il suo nome in un articolo dal titolo *Computing machinery and intelligence*, pubblicato nella rivista di filosofia *Mind* nel 1950 e ripubblicato nel libro *Computers and thought* di Feigenbaum e Feldman [14].

Il programma che emula il dialogo di uno psicoterapeuta è descritto nell'articolo *ELIZA-A computer program for the study of natural language communication between man and machine*, di J. Weizenbaum, pubblicato nella rivista *Communications of the Association for Computing Machinery* nel 1966 [15].

Il programma del robot del mondo a blocchi è descritto nella tesi di dottorato di Terry Winograd, pubblicata come libro dal titolo *Understanding natural language* (Academic Press, New York, 1972) [16].

Il programma che ha generato la storia di Truman e Horace è descritto nell'articolo *A planning mechanism for generating story text*, di Tony Smith e Ian Witten, pubblicato negli atti della *10th International Conference on Computing and the Humanities* nel 1990 [13].

Bibliografia

- [1] Nancy Casey and Michael R. Fellows. *This is Mega-Mathematics!* Los Alamos National Labs, 1992. Available at <http://www.c3.lanl.gov/~captors/mega-math>. See also www.ccs3.lanl.gov/mega-math/write.html.
- [2] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, October 1985.
- [3] A. K. Dewdney. *New Turing Omnibus*. W. H. Freeman & Co., New York, NY, USA, 1993.
- [4] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
- [5] D. Harel. *Algorithmics - The Spirit of Computing*. Addison-Wesley, 2004.
- [6] John Haugeland. *Artificial Intelligence: The Very Idea*. Massachusetts Institute of Technology, Cambridge, MA, USA, 1985.
- [7] Jeff Johnson. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010.
- [8] David A. Klarner. *The Mathematical gardner*. Prindle, Weber and Schmidt Boston, 1981.
- [9] Marek Kubale, editor. *Graph Coloring*. American Mathematical Society, 2004.
- [10] Donald A. Norman. *The Design of Everyday Things*. Basic Books, Inc., New York, NY, USA, 2002.
- [11] Dorothy Elizabeth Robling Denning. *Cryptography and Data Security*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1982.
- [12] Bruce Schneier. *Applied Cryptography (2Nd Ed.): Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., New York, NY, USA, 1995.
- [13] Tony C. Smith and Ian H. Witten. A planning mechanism for generating story texts. *Literary and Linguistic Computation*, 6(2):119–126, 1991.
- [14] A. M. Turing. Computers & thought. chapter Computing Machinery and Intelligence, pages 11–35. MIT Press, Cambridge, MA, USA, 1995.

- [15] Joseph Weizenbaum. Eliza – a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, January 1966.
- [16] Terry Winograd. *Understanding Natural Language*. Academic Press, Inc., Orlando, FL, USA, 1972.