

# Olimpiadi di Informatica 2011

## Giornate preparatorie

*Dipartimento di Informatica  
Università di Torino*

marzo 2011

7 - Algoritmi greedy:  
il massimo numero di intervalli disgiunti.  
(versione 04/04/11)

4/4/2011

E. Giovannetti -- OI09.

1

## Il problema.

Dato un insieme di intervalli, in generale non tutti disgiunti fra di loro, trovarne un sottoinsieme costituito da intervalli tutti disgiunti e tale che il numero di intervalli sia il massimo.

Esempio: Il problema *Nimbus* dalle Olimpiadi di Informatica.

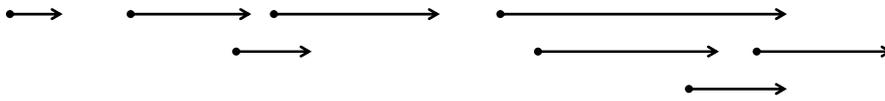
Al celebre maghetto Harry Potter è stata regalata una scopa volante modello Nimbus3000 e tutti i suoi compagni del Grifondoro gli chiedono di poterla provare. Il buon Harry ha promesso che nei giorni a venire soddisferà le richieste di tutti, ma ogni ragazzo è impaziente e vuole provare la scopa il giorno stesso. Ognuno propone ad Harry un intervallo di tempo della giornata durante il quale, essendo libero da lezioni di magia, può fare un giro sulla scopa, e per convincerlo gli offre una fantastica caramella *Tuttigusti+1*. Tenendo presente che una sola persona alla volta può salire sulla Nimbus3000 in ogni istante di tempo, Harry decide di soddisfare, tra tutte le richieste dei ragazzi, quelle che gli procureranno la massima quantità di caramelle (che poi spartirà coi suoi amici Ron e Hermione). Aiutalo a trovare la migliore soluzione possibile.

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

2

### L'algorithmo risolvante.



Scandiamo l'insieme di intervalli: ad ogni passo aggiungiamo all'insieme soluzione un nuovo intervallo, disgiunto da quelli già aggiunti nei passi precedenti.

Secondo quale criterio scegliamo l'intervallo (ovviamente fra quelli disgiunti dai precedentemente inclusi nella soluzione) da aggiungere ogni volta, per fare in modo che alla fine il numero di intervalli sia massimo ?

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

3

### L'algorithmo risolvante.



Proviamo con criteri diversi.

- scegliamo ogni volta l'intervallo **che comincia prima**:  
l'insieme finale ha **4** elementi;

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

4

### L'algorithmo risolvante.



Proviamo con criteri diversi.

- scegliamo ogni volta l'intervallo che comincia prima:  
l'insieme finale ha 4 elementi;
- scegliamo ogni volta l'intervallo **più corto**:  
l'insieme finale ha **3** elementi;

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

5

### L'algorithmo risolvante.



Proviamo con criteri diversi.

- scegliamo ogni volta l'intervallo che comincia prima:  
l'insieme finale ha 4 elementi;
- scegliamo ogni volta l'intervallo più corto:  
l'insieme finale ha 3 elementi;
- scegliamo ogni volta l'intervallo **che finisce prima**:  
l'insieme finale ha **5** elementi.

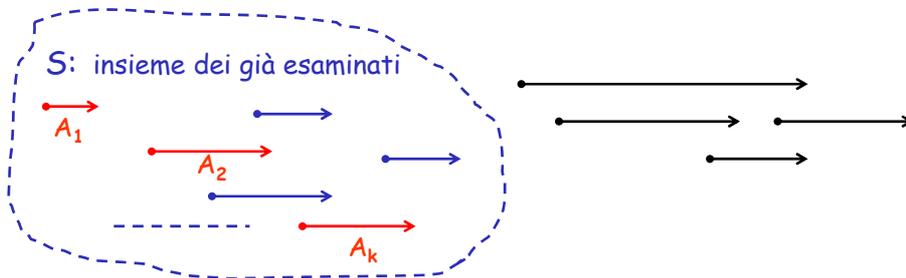
Si dimostra che l'ultimo è effettivamente il criterio corretto, che assicura l'ottimalità della soluzione (cioè che il numero di intervalli scelti sia massimo).

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

6

## Dimostrazione di correttezza: l'invariante del ciclo.



Situazione a un generico passo intermedio (Invariante):

- $S$  è l'insieme di tutti gl'intervalli finora esaminati;
- $A_1, A_2, \dots, A_k$  è la sequenza di intervalli disgiunti già scelti: essa è, per l'insieme  $S$ , una sequenza **massimale**, e inoltre fra le sequenze massimali è **quella che finisce per prima** (cioè quella il cui istante finale dell'ultimo intervallo è il minimo);
- ogni intervallo  $\notin S$  termina dopo tutti gl'intervalli  $\in S$ .

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

7

## Dimostrazione di correttezza: la base.

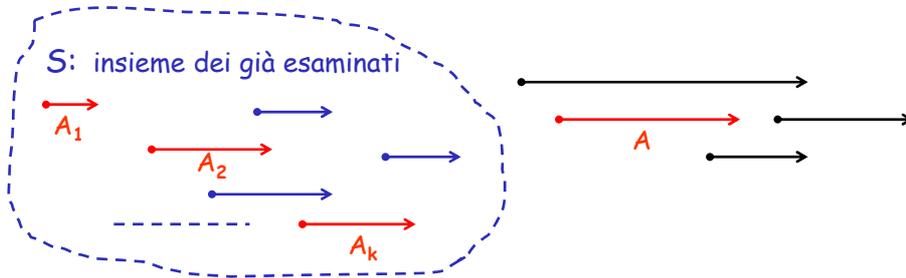
Inizialmente  $S$  è l'insieme vuoto e la sequenza massimale per  $S$  è la sequenza vuota: l'invariante è vacuamente vero.

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

8

## Dimostrazione di correttezza: il passo.



Sia  $A$  l'intervallo che termina prima fra quelli ancora da esaminare (cioè  $\notin S$ ).

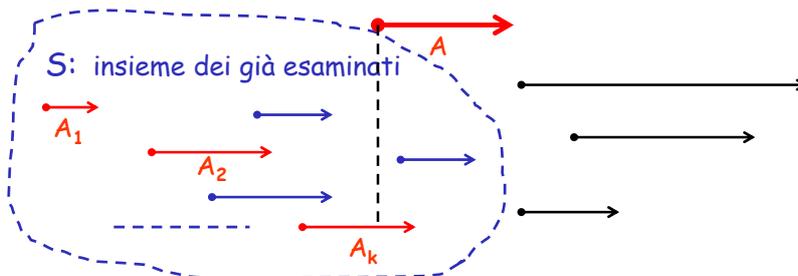
Consideriamo allora l'insieme  $S' = S \cup \{A\}$  e cerchiamo di stabilire qual è per  $S'$  la sequenza massimale che finisce prima.

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

9

## Dimostrazione del passo: caso 1.



caso 1)  $A$  inizia prima della fine della sequenza  $A_1, A_2, \dots, A_k$ .

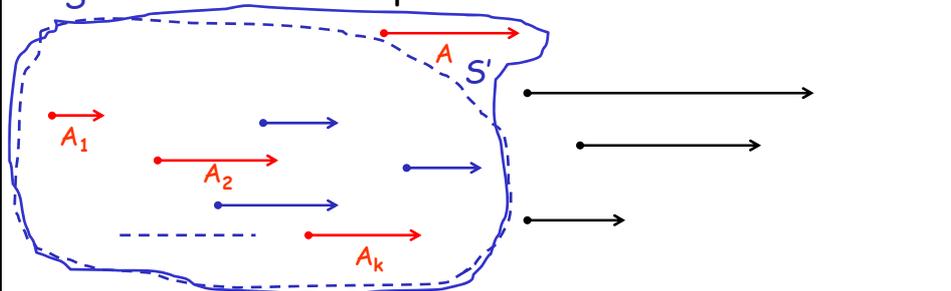
Allora se cerco di costruire una nuova sequenza avente  $A$  come ultimo elemento sono costretto a togliere almeno uno degli elementi  $A_1, A_2, \dots, A_k$ , e quindi non ci guadagno nulla (non aumento il numero di elementi); inoltre la nuova sequenza  $A$  termina dopo ogni elemento di  $S$ , quindi dopo  $A_k$ , quindi sarebbe una sequenza che termina dopo  $A_1, A_2, \dots, A_k$ .

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

10

### Dimostrazione del passo: conclusione del caso 1.



caso 1)  $A$  inizia prima della fine della sequenza  $A_1, A_2, \dots, A_k$ .

Conclusione: per  $S'$  la sequenza massimale che finisce prima è ancora  $A_1, A_2, \dots, A_k$ .

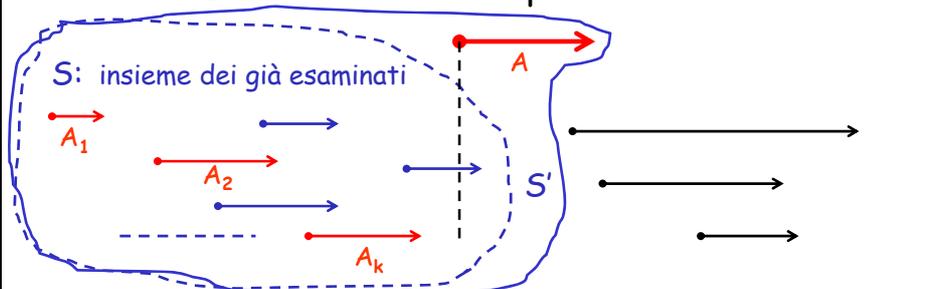
Quindi nel corpo del ciclo l'elemento  $A$  non viene aggiunto alla soluzione ma viene tolto dagli elementi ancora da esaminare (semplicemente incrementando l'indice dell'array degli intervalli, ordinato secondo l'istante finale).

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

11

### Dimostrazione del passo: caso 2.



caso 2)  $A$  inizia dopo la fine della sequenza  $A_1, A_2, \dots, A_k$ .

Allora la sequenza  $A_1, A_2, \dots, A_k, A$ :

- è per  $S'$  una sequenza massimale, perché una seq. massimale per  $S'$  non può avere più di  $k+1$  elementi (altrimenti per  $S$  ci sarebbe una sequenza massimale con più di  $k$  elementi);
- è per  $S'$  la sequenza massimale che finisce prima, perché  $A$  è l'intervallo che finisce prima fra i non appartenenti a  $S$ .

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

12

### Dimostrazione: il terzo punto dell'invariante.

Per come è stato scelto  $A$ , in entrambi i casi ogni intervallo non appartenente a  $S'$  termina non prima di tutti gli intervalli appartenenti a  $S'$ . Quindi anche il terzo punto dell'invariante si conserva.

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

13

### L'algoritmo

- ordina l'insieme degli intervalli in una sequenza  $S$  ordinata secondo l'istante finale (cioè, geometricamente, secondo l'estremo destro del segmento);
- inizializza la soluzione  $Sol$  come sequenza vuota;
- scandisci  $S$  nell'ordine, e per ogni suo elemento  $A$ :
  - se inizia dopo la fine dell'ultimo elemento di  $Sol$ , aggiungilo al fondo di  $Sol$ ;
  - (altrimenti non aggiungerlo).

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

14

## Il codice C++ per il problema Nimbus

```
using namespace std;
struct Intervallo {
    int inizio, fine;
};
int compare(const void *a, const void *b) {
    const Intervallo *pa = (const Intervallo *) a;
    const Intervallo *pb = (const Intervallo *) b;
    return pa->fine - pb->fine;
}
#include <algorithm>
sort(intervalli, intervalli+1000)
int N;
Intervallo intervalli[1000];
```

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

15

## Il codice C++ per il problema Nimbus.

```
int main() {
    ... // input degl'intervalli
    // ordino per istante di fine:
    qsort(intervalli, N, sizeof(Intervallo), compare);
    // in questo problema si chiedeva solo il numero massimo
    // degli intervalli disgiunti, non quali fossero.
    int max = 1;
    int fine = intervalli[0].fine;
    for(int i = 1; i < N; i++) {
        if(intervalli[i].inizio > fine) {
            max++;
            fine = intervalli[i].fine;
        }
    }
    // output max;
}
```

04/04/11 15.55

E. Giovannetti - AlgELab-09-10 - Lez.46

16

## Complessità.

La complessità temporale è quella dell'ordinamento, seguito da una scansione lineare.

Quindi  $O(n \log n) + O(n) = O(n \log n)$ .