

# Olimpiadi di Informatica 2010

## Giornate preparatorie

*Dipartimento di Informatica  
Università di Torino*

marzo 2010  
Parte 3 - Iterazione (es.: ricerca binaria).  
(versione 29/03/10)

3/29/2010

E. Giovannetti -- OI09.

1

## Come si fa a programmare un ciclo ?

Per scrivere un ciclo non banale è indispensabile pensare subito al passo generico, NON al passo iniziale.

Cioè bisogna pensare non agli estremi (inizio o fine del ciclo), bensì ad un generico punto intermedio.

La generica situazione in un punto intermedio, precisata opportunamente, costituisce, come vedremo, ciò che si chiama l'**invariante** del ciclo.



28/02/10 15.36

AlgELab-08-09 - Lez.2

2

Un algoritmo classico, eseguito manualmente da tutti nella vita di ogni giorno:

## La ricerca binaria (o dicotomica).

29/03/10 13.25

E. Giovannetti - Capire 2009

3

### Descrizione a parole, imprecisa, dell'algoritmo.

Precondizione: l'array è ordinato.

- Si confronta l'elemento da cercare con l'elemento centrale dell'array;
- a seconda del risultato del confronto ci si restringe a considerare solo la prima o la seconda metà;
- si confronta l'elemento da cercare con l'elemento centrale della porzione-metà, e di conseguenza ci si restringe a una porzione di lunghezza dimezzata;
- ... e così via ...

Nota Bene: se si cerca di scrivere la procedura basandosi direttamente su tale descrizione, si ottiene una procedura errata e/o brutta !

29/03/10 13.25

E. Giovannetti - Capire 2009

4

## Situazione al passo generico, o INVARIANTE.

Dalla descrizione precedente, si ricava che:  
 ad un generico passo intermedio si ha una porzione di array  
 alla quale si è ristretta la ricerca.  
 Tale porzione non necessariamente si trova all'inizio o alla  
 fine dell'array; in generale si troverà all'interno dell'array:

Per individuarla sono quindi necessari due indici.



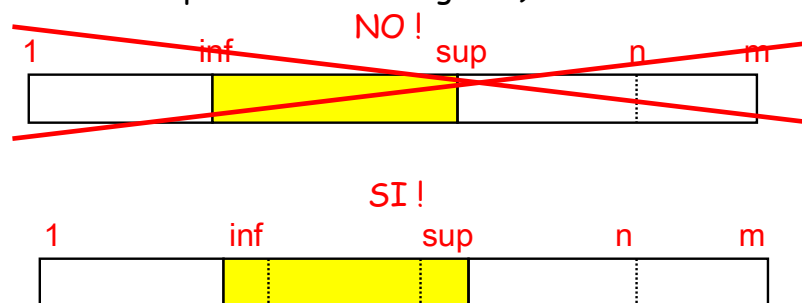
29/03/10 13.25

E. Giovannetti - Capire 2009

5

## Attenzione !

Quando, nella rappresentazione grafica di un array, si indica  
 la posizione di un indice, tale indice deve essere scritto  
 NON in corrispondenza della linea verticale di separazione  
 fra due parti dell'array o della linea iniziale o della linea  
 finale, bensì in corrispondenza di una cella dell'array (che  
 poi, se è quella iniziale o finale di una porzione, può anche  
 non essere esplicitamente disegnata).



29/03/10 13.25

E. Giovannetti - Capire 2009

6

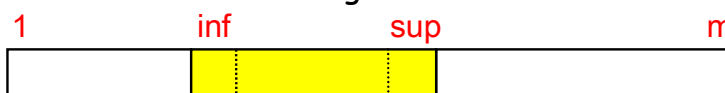
Inoltre ...

Quando si deve indicare la prima o l'ultima cella di una porzione significativa di array, è più chiaro tracciare una linea tratteggiata invece di una linea continua, oppure non tracciarla affatto (a seconda di cosa si vuol evidenziare).

Piuttosto che



meglio



29/03/10 13.25

E. Giovannetti - Capire 2009

7

## Ricerca binaria in array ordinato (in pseudo-Pascal)

SITUAZIONE AL PASSO GENERICO



Il valore  $x$  da cercare, se è presente nell'array, si trova nella porzione di array compresa fra gl'indici **inf** e **sup** (inclusi).

PASSO GENERICO

Calcolo il valore dell'indice  $i$  dell'elemento centrale della porzione di array; confronto  $x$  con tale elemento  $a[i]$ ; sono possibili tre casi:

- $x < a[i]$ :  $x$ , se c'è, si trova nella porzione compresa fra **inf** e  **$i-1$**  (inclusi);
- $x > a[i]$ :  $x$ , se c'è, si trova nella porzione compresa fra  **$i+1$**  e **sup** (inclusi);
- $x = a[i]$ :  $x$  c'è, e si trova nella posizione  **$i$** .

29/03/10 13.25

E. Giovannetti - Capire 2009

8

## Ricerca binaria in array ordinato (in pseudo-Pascal)

### SITUAZIONE AL PASSO GENERICO



Come faccio a calcolare l'indice  $i$  ?

Attenzione:

Se penso alla situazione iniziale, in cui la parte "gialla" è tutto l'array,  $i$  è semplicemente  $n \text{ div } 2$ .

Ma NON bisogna pensare alla situazione iniziale !  
Bisogna mettersi mentalmente al passo generico !

Che cosa è  $i$  in tal caso ?  $(\text{sup} - \text{inf})/2$  ?

No ! Supponiamo  $\text{inf} = 10$  e  $\text{sup} = 20$ : qual'è il punto medio ?  
È la media aritmetica:  $(10 + 20)/2 = 15$ .

29/03/10 13.25

E. Giovannetti - Capire 2009

9

### Casi 1 e 2



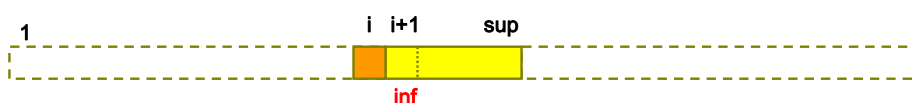
•  $x < a[i]$ : l'elemento  $x$ , se c'è, si trova nella porzione compresa fra  $\text{inf}$  e  $i-1$  (inclusi): quindi

```
if x < a[i] then sup := i-1;
```



•  $x > a[i]$ : l'elemento  $x$ , se c'è, si trova nella porzione compresa fra  $i+1$  e  $\text{sup}$  (inclusi): quindi

```
else if x > a[i] then inf := i+1;
```



29/03/10 13.25

E. Giovannetti - Capire 2009

10

### Terzo caso

- `x = a[i]` ho trovato il valore cercato, quindi termino la funzione dando come risultato il valore dell' indice `i`  
*else dà come risultato i e termina*



29/03/10 13.25

E. Giovannetti - Capire 2009

11

### Qual è la condizione per cui il ciclo deve continuare?

La porzione di array su cui fare la ricerca non deve essere vuota; cioè  $inf \leq sup$

NOTA: se  $inf = sup$  la porzione non è vuota, ma contiene un elemento.

### Quali sono i valori iniziali di inf e sup ?

Inizialmente la porzione di array su cui effettuare la ricerca è l'intero array.

$inf := 1;$        $sup := \text{indice dell'ultimo elemento};$

**Che cosa succede se il ciclo termina normalmente, cioè senza interrompersi per dare il risultato ?**

Vuol dire che l'elemento cercato non c'è.

29/03/10 13.25

E. Giovannetti - Capire 2009

12

Se l'elemento cercato non c'è, il ciclo termina sempre ?

- Osserva: a ogni ripetizione del ciclo l'intervallo di ricerca si restringe almeno di un elemento, perché viene escluso l'elemento centrale.
- Quindi, se l'elemento cercato non viene trovato, prima o poi i due indici inf e sup si incrociano, e il ciclo termina.

29/03/10 13.25

E. Giovannetti - Capire 2009

13

Ricerca binaria con risultato l'indice (in pseudo-Pascal)

```
function ricBin(x:integer; a:array): integer;
var inf, sup, i: integer;
begin

    i := (inf + sup) div 2;
    if x < a[i] then sup := i-1
    else if x > a[i] then inf := i+1
    else dà_come_risultato i e termina

end;
```

29/03/10 13.25

E. Giovannetti - Capire 2009

14

## Ricerca binaria con risultato l'indice (in pseudo-Pascal)

```

function ricBin(x:integer; a:array): integer;
var inf, sup, i: integer;
begin

  while inf <= sup do begin
    i := (inf + sup) div 2;
    if x < a[i] then sup := i-1
    else if x > a[i] then inf := i+1
    else dà_come_risultato i e termina
    end;
    dà_come_risultato -1
  end;
end;

```

Nota: Stabiliamo che il risultato **-1**, che rappresenta una posizione inesistente nell'array, indichi che l'elemento cercato non è contenuto nell'array.

29/03/10 13.25

E. Giovannetti - Capire 2009

15

## Ricerca binaria con risultato l'indice (in pseudo-Pascal)

```

function ricBin(x:integer; a:array): integer;
var inf, sup, i: integer;
begin
  inf := 1; sup := lunghezza di a;
  while inf <= sup do begin
    i := (inf + sup) div 2;
    if x < a[i] then sup := i-1
    else if x > a[i] then inf := i+1
    else dà_come_risultato i e termina
    end;
    dà_come_risultato -1 e termina
  end;
end;

```

Nota: Stabiliamo che il risultato **-1**, che rappresenta una posizione inesistente nell'array, indichi che l'elemento cercato non è contenuto nell'array.

29/03/10 13.25

E. Giovannetti - Capire 2009

16



## Codifica in Dev-Pascal/FreePascal

```

const maxlun = 1000000;
type arraytype = array[1..maxlun] of integer;

function ricercaBin(x: integer; var a: arraytype;
                   n: integer): integer;
var inf, sup, i: integer;
begin
  inf:= 1;  sup := n;
  while(inf <= sup) do begin
    i := (inf + sup) div 2;
    if(x < a[i]) then sup := i -1
    else if(x > a[i]) then inf := i +1
    else exit(i);
  end;
  exit(-1);
end;

```

29/03/10 13.25

E. Giovannetti - Capire 2009

17

Codifica in Dev-Pascal/FreePascal con array dinamici,  
indiciati a partire da 0 come in tutti i linguaggi moderni

```

function ricBin(x: integer; var a: array of integer):
integer;
var inf, sup, i: integer;
begin
  inf:= 0;  sup := high(a);
  while(inf <= sup) do begin
    i := (inf + sup) div 2;
    if(x < a[i]) then sup := i -1
    else if(x > a[i]) then inf := i +1
    else exit(i);
  end;
  exit(-1);
end;

```

29/03/10 13.25

E. Giovannetti - Capire 2009

18

## Versione C

```
int ricercaBin(int x, int a[], int n) {  
    int inf = 0, sup = n-1;  
    while(inf <= sup) {  
        int i = (inf + sup)/2;  
        if(x < a[i]) sup = i-1;  
        else if(x > a[i]) inf = i+1;  
        else return i;  
    }  
    return -1;  
}
```

## Nota:

In quasi tutti i linguaggi moderni gli array sono indicizzati a partire da 0 invece che da 1; quindi in un array di lunghezza  $n$ , cioè contenente  $n$  elementi, gli indici vanno da 0 a  $n-1$  (inclusi).