

## La ricerca in un vettore

### Il problema

Si supponga di avere un insieme di  $N$  elementi di tipo *Integral* (interi o caratteri), rappresentato da un vettore  $V(i)$ , ( $i=1, N$ ).

Considerato un elemento  $K$ , si vuole determinare se tale elemento, denominata **chiave**, coincide con uno degli elementi dell'insieme precedente, cioè se esso è uguale ad uno degli elementi del vettore  $V$ .

Il problema enunciato ricorre in tutti i casi di ricerca in un archivio, elenco, lista, ecc.

### L'analisi

La ricerca dell'eventuale uguaglianza di  $K$  ad uno degli elementi di  $V$  è basata, nel seguito, sul metodo dei confronti.

I metodi esaminati sono due:

- la ricerca sequenziale o lineare;
- la ricerca binaria o dicotomica o logaritmica.

### La ricerca sequenziale

Si confronta ripetutamente la chiave  $K$  con ciascuno degli elementi del vettore, finché non si trova  $V(i) = K$  (*ricerca con successo*) oppure finché non sono stati considerati tutti gli elementi del vettore senza trovarne nessuno uguale a  $K$  (*ricerca senza successo*).

Due valutazioni diverse a seconda dell'esito della ricerca:

- ricerca con successo: si fa riferimento al numero medio di confronti, che si ottiene dividendo il numero totale di confronti necessari per ricercare tutti gli elementi per il numero degli elementi stessi. Siccome per individuare il primo elemento si effettua un confronto, per il secondo due e così via, il numero totale di confronti è :  $1+2+3+4+5+.... N = N(N+1)/2$   
Il numero medio di confronti risulta  $(N+1)/2$ ;
- ricerca senza successo: l'algoritmo esamina sempre tutto il vettore, quindi il numero di confronti è sempre  $N$ . Il numero di confronti in caso di ricerca senza successo è lo stesso che nella ricerca con successo  $N(N+1)/2$ , perché si procede alla scansione del vettore finché  $K \leq V[i]$ . A questo punto o  $K = V[i]$ , e la ricerca è con successo, oppure si possono interrompere i confronti, e la ricerca risulta senza successo. Pertanto si determina il numero medio di confronti come in caso di successo.

### La ricerca binaria

Si basa sul requisito che gli elementi di  $V$  siano ordinati in ordine crescente (o decrescente), così che:

$$V(1) \leq V(2) \leq V(3) \leq ..... \leq V(N-1) \leq V(N)$$

Nel seguito si considera che l'ordine sia crescente.

Si vuole determinare a quale elemento  $V(i)$  è eventualmente uguale l'elemento  $K$ .

Si confronta la chiave  $K$  con l'elemento che si trova a metà del vettore. Se l'elemento individuato non è uguale a quello cercato si prosegue la ricerca nel semivettore inferiore o superiore, a seconda che la chiave  $K$  sia più piccola o più grande dell'elemento che si trova a metà.

Il procedimento continua iterativamente in modo da suddividere i semivettori via via individuati.

La ricerca termina con successo quando l'elemento  $V[i]$  considerato ad un certo passo è proprio uguale a  $K$ . La ricerca termina con insuccesso quando la parte di vettore considerata è costituita da un solo elemento.

L'algoritmo è il più veloce tra quelli di ricerca basati sul confronto di chiavi.  
In un vettore di dimensione

$$N = 2^h - 1$$

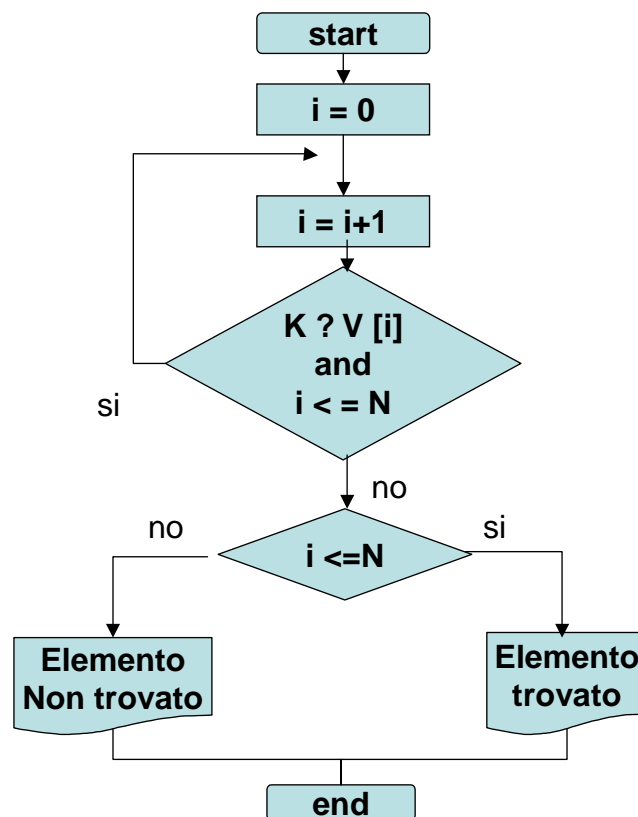
l'algoritmo deve compiere

$$h = \log_2(N+1)$$

passi (e quindi confronti) per la ricerca senza successo, mentre i confronti possono essere di meno per la ricerca con successo.

## Il progetto

La figura che segue riporta il flow-chart dell'*algoritmo di ricerca sequenziale*.



La figura che segue riporta il flow-chart dell'*algoritmo di ricerca binaria*.

