

LA RICERCA SEQUENZIALE

Supponiamo di voler calcolare la media di un milione di numeri.

Possiamo immaginare che i valori siano stati registrati su un archivio e che vengano letti uno dietro l'altro. L'elaboratore man mano li somma e tiene conto di quanti sono. Alla fine non fa altro che dividere il totale ottenuto per il numero dei dati. il tempo impiegato per effettuare questa operazione sarà di qualche secondo. Dato che i dati sono memorizzati su disco, che devono essere letti nella memoria centrale e che devono essere letti tutti, il tempo considerato è accettabile.

Il metodo di *scansione dei dati* effettuato è inevitabile nel caso si debba calcolare la media di un insieme di numeri.

Un tale metodo può essere utilizzato anche quando si cerca un dato in un insieme non ordinato.

È il metodo più elementare per ricercare un dato in una qualunque sequenza. Quando viene ricercato ad esempio un nome in una lista registrata nella memoria di un computer, *nel peggiore dei casi* (che può capitare molto spesso, per esempio se il dato richiesto non c'è), si dovranno esaminare tutti i dati della collezione, uno alla volta: un'operazione **il cui costo dipende dal numero dei dati**.

Il metodo di ricerca dei dati effettuato con questa modalità viene detto *metodo di ricerca sequenziale*.

È ovvio che il tempo necessario per accedere ad una informazione qualsiasi, soprattutto per archivi di grosse dimensioni, sia notevole; è dunque necessario in questi casi usare metodi di ricerca più efficienti.

LA RICERCA SEQUENZIALE

Un esempio di archivio di grosse dimensioni può essere quello che registra i dati anagrafici di tutti i cittadini di un comune. con il metodo di ricerca sequenziale la procedura consisterebbe nel controllare uno dopo l'altro tutti i cittadini registrati fino a trovare quello desiderato, con tempi di accesso all'informazione inaccettabili. Un tale tipo di ricerca andrebbe bene se il cittadino fosse uno dei primi dell'elenco, ma non altrettanto bene se si trovasse in fondo all'elenco. Se i cittadini sono n , il tempo di ricerca per individuare un qualsiasi cittadino sarebbe *mediamente proporzionale* a $n/2$.

Si pone allora il problema di utilizzare un metodo di ricerca che sia nettamente più efficiente.

Considerando che l'elenco dell'anagrafe è ordinato alfabeticamente, l'idea fondamentale del metodo è quella di cominciare la ricerca non dal primo nome dell'elenco (cioè dalla lettera A), ma da quello centrale, cioè a metà elenco. A questo punto si confronta questo elemento con l'oggetto della nostra ricerca: in questo modo: se dobbiamo cercare un nome che sta nella seconda metà della lista, si elimina dalla ricerca in maniera automatica la prima metà. La ricerca continua poi nello stesso modo, eliminando di volta in volta metà dell'elenco rimasto, fino all'ultimo confronto che ci darà l'informazione richiesta.

Un metodo comunemente usato per la ricerca di un nome in un vocabolario è quello che segue: apriamo il dizionario in qualunque punto e quindi andiamo avanti o indietro a seconda della parola cercata, visto che il dizionario è ordinato alfabeticamente

Questo modo di procedere è detto “**ricerca binaria**” ed è uno dei migliori che si conoscano.

Se si pensa ad un elenco con 1.000.000 di elementi, già il primo confronto riduce il numero di elementi ancora da considerare a soli 500.000. Il secondo confronto li riduce a 250.000 e il terzo a 125.000. Continuando a dividere per 2, si vede facilmente che dopo soli 19 confronti ci si è ridotti a un unico elemento. Il 20° confronto pertanto ci dirà se l'elemento che cerchiamo è presente nell'elenco oppure no.

In questo caso si passa da 500.000 confronti, necessari in media per la ricerca sequenziale, a 20 confronti per la ricerca binaria.

Parlando di tempi assoluti, tutto ciò si traduce in un risparmio di 100 volte nella peggiore delle ipotesi.

Supponiamo, per semplificare, di volere cercare un numero in un vettore ordinato di numeri.

Nella ricerca binaria, ad ogni passo dimezziamo la dimensione del vettore in cui cercare il numero, pertanto se il vettore è di dimensione n , nel peggiore dei casi dovremo **analizzare solo $\log_2 n$ elementi**.

Osserviamo che $\log_2 n$ ci dice il numero di volte che n si deve dividere per due per arrivare ad 1.

\

Riassumendo, se abbiamo una collezione di $n = 8000$ oggetti, una **ricerca sequenziale**, nel peggiore dei casi deve visitarli tutti e, quindi, farà **n confronti** mentre la **ricerca binaria** fa:

8000, 4000, 2000, 1000, 500, 250, 125, 62, 31, 15, 7, 3, 1,

13 confronti, ovvero

$\log_2 8000 = 13$