INFORMATICA GENERALE - I-Z

Claudia Malvenuto - Daniele A. Gewurz
Scheda esercizi n. 4

1. (a) Trovare un invariante di ciclo per la ricerca lineare e usarlo per dimostrare la correttezza dell'algoritmo.
   (b) Ripetere il punto precedente per la ricerca dicotomica.
2. Calcolare il tempo di esecuzione dell'*insertion sort* nel caso medio.
3. Scrivere un invariante di ciclo per l'algoritmo di *selection sort*.
4. Consideriamo l'algoritmo di ordinamento *bubble sort* ("ordinamento a bolle"). Data una lista A di n numeri, questo algoritmo comincia confrontando $A[1]$ e $A[2]$ e scambiandoli di posto se $A[1] > A[2]$; poi passa a $A[2]$ e $A[3]$ etc., fino ad $A[n-1]$ e $A[n]$. Se in questo primo passaggio è stato effettuato almeno uno scambio, si svolge un nuovo passaggio ricominciando da $A[1]$ e $A[2]$, e così via fino a un passaggio in cui non si siano svolti scambi.
   (a) Scrivere una descrizione in pseudocodice del *bubble sort*.
   (b) Migliorare l'algoritmo tenendo conto del fatto che dopo l'i-ma iterazione almeno gli ultimi i elementi sono al loro posto definitivo.
   (c) Descrivere un invariante di ciclo per questa variante e dimostrare la correttezza dell'algoritmo.
   (d) Calcolare la complessità dell'algoritmo, nei casi migliore, peggiore e medio.
   (e) Considerare una variante dell'algoritmo in cui l'unica differenza consiste nel cominciare da $A[n-1]$ e $A[n]$ e procedere da destra a sinistra. (Questa variante può essere utile per esempio se si aggiungono nuovi elementi in coda a una lista già ordinata.)
   (f) Come il punto (b), ma tenendo conto che dopo ogni iterazione gli elementi al posto definitivo sono in realtà tutti quelli in una posizione successiva all'ultimo scambio.
5. Siano data una lista non ordinata di $2n$ numeri e la si voglia ripartire in due liste L_1 e L_2 di n numeri ognuna, tali che la differenza tra la somma degli elementi di L_1 e la somma di quelli di L_2 sia maggiore possibile. Descrivere una strategia per farlo in un tempo $O(n \log n)$.
6. Dimostrare che, se si devono fondere due sequenze ordinate di lunghezza n ciascuna, è possibile che siano necessari $2n-1$ confronti.

   (Suggerimento: Considerare il caso in cui, se le sequenze sono $a_1 < a_2 < \ldots < a_n$ e $b_1 < b_2 < \ldots < b_n$, ogni volta che si confrontano a_i e b_j si osserva che $a_i < b_j$ se e solo se $i \leq j$.)
7. Supponiamo di avere due liste M e N di numeri, di lunghezza rispettivamente m e n (con $m < n$), e di voler determinare se hanno elementi in comune. Scrivere tre algoritmi per risolvere questo problema, seguendo le seguenti strategie:
   (a) ordinare (con un algoritmo che impiega un tempo $O(n \log n)$) la lista N e poi, per ogni elemento x della lista M, eseguire una ricerca binaria di x in N;
   (b) come il punto precedente, ma invertendo i ruoli di M e N;

(c) ordinare entrambe le liste e poi cercare eventuali elementi in comune così: se il minimo elemento di una lista è uguale al minimo dell'altra abbiamo finito; altrimenti scartiamo il più piccolo tra i due e proseguiamo ricorsivamente.

Analizzare il tempo di esecuzione dei tre algoritmi (in funzione di m e n) e decidere qual è il più veloce.

8. Un algoritmo di ordinamento è detto *stabile* se, quando sono presenti più elementi uguali, lascia immutato il loro ordine relativo.

   Quali degli algoritmi di ordinamento studiati sono stabili? Notate che in alcuni casi l'eventuale stabilità non è una proprietà intrinseca dell'algoritmo, ma dipende dall'implementazione.

   La stabilità è importante nella pratica: per esempio, se in un database si hanno i dati relativi a varie persone (nome, cognome, data di nascita etc.), ordinati alfabeticamente per cognome, e li si vuole riordinare per data di nascita, allora se l'algoritmo è stabile, all'interno delle persone nate lo stesso giorno viene mantenuto l'ordine alfabetico.

9. Un certo computer impiega in media circa 2 centesimi di secondo per ordinare con il *merge sort* (o con un altro algoritmo che richiede un tempo $O(n \log n)$) un elenco di 10.000 numeri dati in ordine casuale. Quanto impiegherà per ordinarne 100.000?

10. Svolgere i seguenti esercizi sulla ricorsione: 4.4-1, 4.4-2, 4.4-5 (Cormen, pag. 78); 4-1, 4-3 (Cormen, pag. 90).