

Algoritmi e Strutture Dati

4 Febbraio 2004

1. *Tempo disponibile 180 minuti (è ammesso ritirarsi entro 90 minuti)*
2. *Sono ammessi al più 3 scritti consegnati per A.A.*
3. *Non è possibile consultare appunti, libri o persone, né uscire dall'aula*
4. *Nella valutazione dello scritto ogni esercizio conta 6 punti (e quindi si raggiunge 18 con 3 esercizi risolti correttamente e 30 con 5 esercizi risolti correttamente)*
5. *Le soluzioni degli esercizi devono:*
 - a. *Spiegare a parole l'algoritmo usato (anche con eventuali disegni)*
 - b. *commentare l'eventuale procedura Pascal (dettagliando il significato delle variabili)*
 - c. *giustificare la correttezza e tutti i passaggi matematici*
 - d. *dimostrare la complessità (con equazioni di ricorrenza se necessario)*

1. Si scriva una procedura Pascal che riceva come parametro d'ingresso un vettore di n elementi, tale che il suo tempo di esecuzione $T(n)$ verifichi la ricorrenza:

$$T(n) = c \quad \text{per } n < 5$$

$$T(n) = 4T(n/2) + dn^2 \quad \text{altrimenti}$$

con c e d costanti. Si calcoli anche l'ordine di grandezza di $T(n)$.

2. Dato un vettore contenente n elementi interi distinti, si vuole decidere se esistono due elementi che danno per somma 13. Si assuma che il vettore possa essere ordinato oppure no, e che gli elementi possano essere tutti positivi oppure no. Si scrivano algoritmi efficienti (e possibilmente di complessità ottima), considerando ciascuno dei quattro casi possibili.

3. Data una lista L di interi, si vuole modificarla cancellando tutti gli elementi i cui valori compaiono esattamente una volta, mantenendo lo stesso ordine che gli elementi avevano inizialmente (p.e. se l'ingresso è $L = 6, 1, 6, 4, 1, 1, 3, 5$ allora il risultato è $L = 6, 1, 6, 1, 1$). Si scriva una procedura Pascal *utilizzando gli operatori* per le liste visti a lezione.

4. Si scriva la procedura Pascal *Depth-First Search (DFS)* vista a lezione. Si esegua la procedura sul grafo non orientato $G = (N, A)$, $N = \{1, 2, 3, 4, 5\}$, $A = \{[1,2], [1,3], [1,5], [2,5], [3,4], [3,5]\}$ a partire dal nodo 3, assumendo che i vettori di adiacenza siano ordinati in modo *crescente* e mostrando il contenuto dei vettori di adiacenza.

5. Dato un albero binario T contenente elementi interi, si vuole modificarlo aggiungendo un figlio ad ogni foglia che contiene un elemento pari. Si scriva una procedura Pascal di complessità ottima assumendo che l'albero sia *realizzato con puntatori*.

6. Siano date due stringhe $P = p_1p_2\cdots p_m$ e $T = t_1t_2\cdots t_n$ di caratteri alfabetici. Si progetti un algoritmo di "programmazione dinamica" per individuare la più lunga *sottostringa* comune tra P e T (per esempio, se $P = 9, \underline{15}, \underline{6}, \underline{2}, \underline{5}, 10, 3$ e $T = 8, 2, \underline{15}, \underline{6}, \underline{2}, \underline{5}, 9, 10, 3, 1$ allora il risultato è: 15, 6, 2, 5).