

Laurea in “Informatica”
Corso di “Algoritmi e Strutture Dati”
14 Luglio 2009

1. Tempo disponibile 180 minuti. È ammesso ritirarsi entro 90 minuti.
2. Sono ammessi al più 3 scritti consegnati per l'A.A. 2008/09 (Giugno 2009 - Febbraio 2010)
3. Non è possibile consultare appunti, libri o persone, né uscire dall'aula.
4. 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. Data una lista L di interi, si vuole modificarla duplicando tutti gli elementi i cui valori compaiono esattamente una volta e cancellando tutti gli altri elementi, mantenendo lo stesso ordine che gli elementi avevano inizialmente (p.e. se l'ingresso è $L = 6, 2, 6, 4, 5, 3, 6, 4$ allora il risultato è $L = 2, 2, 4, 4, 3, 3$). Si scriva una procedura Pascal *utilizzando gli operatori* per le liste visti a lezione.

2. Siano dati un albero binario T contenente elementi interi positivi ed un intero k . Si vuole modificare T aggiungendo un figlio destro ad ogni foglia per la quale la somma degli elementi contenuti nel percorso dalla radice alla foglia sia uguale a k e in modo che la somma degli elementi nel percorso dalla radice alla nuova foglia inserita sia $k + 2$. Si scriva una procedura Pascal di complessità ottima assumendo che l'albero sia *realizzato con puntatori*.

3. Si supponga di memorizzare gli insiemi (di interi distinti) per mezzo di alberi bilanciati di ricerca. Come si realizzano efficientemente le operazioni UNIONE, INTERSEZIONE e DIFFERENZA e quali sono le loro complessità?

4. Si scriva una procedura Pascal ricorsiva che riceva in input un vettore di n interi tale che la sua complessità $T(n)$ verifichi la relazione di ricorrenza seguente:

$$T(n) = d, \text{ se } n < 17,$$

$$T(n) = 3T(n/2) + 2T(n-1) + cn, \text{ altrimenti,}$$

con d e c costanti. L'ordine di grandezza di $T(n)$ è polinomiale oppure no?

5. Si scriva un algoritmo *non deterministico* di complessità $O(1)$ per risolvere il problema del Sudoku (in tale problema, è data in input una tabella con 9 righe e 9 colonne, partizionata in 9 sottotabelle ciascuna di 3 righe e 3 colonne; inizialmente alcune caselle della tabella contengono interi compresi tra 1 e 9, mentre le altre caselle sono vuote; si vuole riempire le caselle vuote in modo che in ogni riga, in ogni colonna ed in ogni sottotabella compaiano una e una sola volta tutti i numeri da 1 a 9).

6. Si descriva a parole l'euristica LPT vista a lezione e la si esegua (a mano) per risolvere il problema dello SCHEDULING MULTIPROCESSORE col seguente input: 3 processori, 8 programmi, e tempi di esecuzione dei programmi uguali a 4, 7, 3, 5, 1, 6, 2, 4.