

Laurea in “Informatica”

Corso di “Algoritmi e Strutture Dati”

19 Febbraio 2008

1. Tempo disponibile 180 minuti. È ammesso ritirarsi entro 90 minuti.
2. Si possono consegnare al più 3 scritti tra Giugno 2007 e Febbraio 2008.
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. Si valuti l'ordine di grandezza della complessità $T(n)$ della seguente funzione Pascal:

```
function PIPPO(n: integer): integer;  
  var i, j: integer;  
  begin  
    for i := 0 to 1 do j := 2;  
    if n > 10 then  
      PIPPO := 5* PIPPO(n div 2) + PIPPO(n div j) + j*n div 2  
    else if n > 2 then  
      PIPPO := 5* PIPPO(n div 2) + PIPPO(n - j) + j*n div 2  
    else  
      PIPPO := 1  
  end;
```

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

CONTINUA SULL'ALTRA FACCIATA

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 $L = 6, 1, 6, \underline{4}, 1, 1, \underline{3}, \underline{5}$ 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 riscriva la procedura Pascal *Depth-First Search (DFS)* vista a lezione e la si esegua sul grafo non orientato $G=(N,A)$, $N=\{1,2,3,4,5\}$, $A=\{[1,2],[1,3],[1,5],[3,5],[3,4],[4,1],[4,2]\}$ a partire dal nodo 5. Si disegni il grafo e si mostrino la memorizzazione di G con vettori di adiacenza e l'ordine di visita dei nodi e degli archi (assumendo che i vettori di adiacenza siano ordinati in modo *crescente*).

5. Si indichi il contenuto di una tabella hash di dimensione 13, inizialmente vuota, dopo l'inserzione, nell'ordine, delle chiavi: Q, U, E, S, T, I, A, P, P, E, L, L, I. Si usi la funzione hash $H(k) = k \bmod 13$ per la k -esima lettera dell'alfabeto italiano e il metodo di scansione lineare a passo unitario. Si indichi successivamente il contenuto della tabella dopo avervi cancellato, nell'ordine: B, A, S, T, A, e poi inserito, nell'ordine: U, L, T, I, M, O.

6. Si scriva la procedura HEAPSORT vista a lezione e la si esegua (a mano) per ordinare alfabeticamente gli 11 elementi: O, R, D, I, N, A, M, E, N, T, O. Si illustri con disegni, passo dopo passo, il contenuto dello heap durante l'esecuzione.