Laurea in "Informatica"

Corso di "Algoritmi e Strutture Dati"

4 Giugno 2008

1. *Tempo disponibile 180 minuti. È ammesso ritirarsi entro 90 minuti.*
2. *Si possono consegnare al più 3 scritti tra Giugno 2008 e Febbraio 2009.*
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 MISTERO(n: integer): integer;
    var i, j: integer;
    begin
        for i := 0 to 1 do j := 2;
        if n > 10 then
            MISTERO := 5*MISTERO(n div 2) + MISTERO(n div j) + j*n
div 2
        else if n > 2 then
            MISTERO := 5*MISTERO(n div 2) + MISTERO(n - j) + j*n div
2
        else
            MISTERO := 1
    end;
```

2. Sia dato un albero binario A implementato con puntatori, in cui ogni nodo contiene un elemento intero. Scrivere una funzione Pascal di complessità ottima per cancellare ogni foglia di A che contiene un elemento maggiore di quello contenuto nel padre.

3. Utilizzando gli operatori per le liste visti a lezione, si scriva una procedura Pascal di complessità *ottima* che, data una lista L di interi, modifica L e restituisce una seconda lista M contenenti, rispettivamente, tutti gli elementi dispari e tutti quelli pari, rispettando l'ordine che avevano inizialmente in L. (p.e., se in input L = 3, 7, 8, 1, 4, allora si ottengono in output L = 3, 7, 1 ed M = 8, 4).

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 3. 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. Dati un vettore ordinato di n interi distinti e due interi s e d, con $s<d$, si vuole decidere se esistono o no elementi del vettore che appartengono all'intervallo $[s, d]$. Si scriva una procedura (o funzione) Pascal *divide et impera* modificando la *ricerca binaria* vista a lezione.

6. Dati un insieme A di n interi ed un intero k, si vuole decidere se esiste una partizione di A in tre sottoinsiemi B, C e D tali che il prodotto degli elementi di B sia k e la somma degli elementi di C

sia uguale a k volte la somma degli elementi di D. Si scriva una procedura Pascal *non deterministica* di complessità polinomiale.