

# Algoritmi e Strutture Dati con Laboratorio (modulo I)

Testo della prova scritta del 14 febbraio 2012

docente: Luciano Gualà

Cognome:..... Nome:..... Matr.:..... Corso di Laurea:.....

## Esercizio 1 [10 punti]

(a) Si ordinino le seguenti funzioni in ordine non decrescente di tasso di crescita asintotica.

Per ogni coppia di funzioni  $f_i(n), f_{i+1}(n)$  adiacenti nell'ordinamento si specifichi se  $f_i(n) = \Theta(f_{i+1}(n))$  o se  $f_i(n) = o(f_{i+1}(n))$ .

Le funzioni sono:  $2^n$ ,  $\frac{n^{0.52}+1}{n^{0.01} \log^{10} n}$ ,  $3^n$ ,  $n \log \log n$ ,  $512\sqrt{n}$ ,  $\frac{n^3}{(n+1)(n+5)}$ ,  $\sqrt{n} \log n$ ,  $2^{n-10}$ ,  $n\sqrt{\log n}$ .

(b) Per un problema sono noti due algoritmi ricorsivi,  $A_1$  e  $A_2$  le cui complessità temporali sono descritte dalle seguenti equazioni di ricorrenza:

$$T_1(n) = 2T_1(n-2) + 1, T_1(1) = 1;$$

$$T_2(n) = 16T_2(n/2) + n^{3.1}\sqrt{n} + n^2 \log^2 n, T_2(1) = 1;$$

Dire, motivando la risposta, quale algoritmo è preferibile usare.

**Esercizio 2 [10 punti]** Sia  $T$  un albero binario di  $n$  nodi con radice  $r$  in cui ogni nodo ha una chiave intera non negativa associata. Si assuma che  $T$  è mantenuto attraverso una struttura collegata e che ogni nodo  $v$  abbia associato i seguenti campi: puntatori al padre e ai figli ( $v.p, v.s, v.d$ ) e chiave del nodo ( $v.ch$ ).

Si progetti un algoritmo con complessità temporale  $O(n)$  che, preso  $T$ , restituisca il numero di nodi di  $T$  che godono della seguente proprietà: la chiave del nodo coincide con la profondità del nodo stesso. Si ricordi che la profondità di un nodo in un albero è pari al livello dell'albero in cui il nodo compare e che la radice ha profondità 0. Si fornisca lo pseudocodice dettagliato dell'algoritmo.

**Esercizio 3 [15 punti]** Un *cammino* è un albero in cui tutti i nodi interni hanno esattamente un figlio. Un *insieme indipendente* di un cammino è un sottoinsieme  $I$  di nodi del cammino tale che per ogni coppia  $u, v \in I$ , non c'è l'arco  $(u, v)$  nel cammino, ovvero  $u$  non è padre di  $v$  e  $v$  non è padre di  $u$ .

Si assuma di avere un cammino di  $n$  nodi e che ad ogni nodo  $v$  del cammino sia associato un valore positivo  $val(v)$ . Si progetti un algoritmo che, preso il cammino, restituisca il valore dell'insieme indipendente di valore massimo, dove il valore di un insieme indipendente  $I$  è definito come  $\sum_{v \in I} val(v)$ . I nodi sono numerati da 1 a  $n$  a partire dalla radice verso l'unica foglia e sono mantenuti in memoria attraverso un vettore  $V$  tale che  $V[i]$  contiene il valore dell' $i$ -esimo nodo. L'algoritmo deve avere complessità temporale  $O(n^3)$ . *Suggerimento*: si usi la programmazione dinamica.