

Algoritmi e Strutture Dati (modulo I)  
Testo della prova scritta del 19 febbraio 2019  
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:  $2^{3n}$ ,  $\frac{n\sqrt[4]{n^8+8}}{n+\sqrt{n+8}}$ ,  $n\sqrt[30]{n^{30}+30}$ ,  $2^n$ ,  $\frac{n^2}{\log \sqrt{n}}$ ,  $\frac{n^2+313}{\sqrt{\log n}}$ ,  $\frac{n^2+\sqrt{n}}{\log n}$ ,  $2^{3\sqrt{\log n}}$ ,  $n^{1.5} \log^{3.5} n$ .

- (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:

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

$$T_2(n) = 49T_2(n/7) + n \log^2 n, T_2(1) = 1;$$

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

**Esercizio 2 [12 punti]** Sia  $T$  un albero binario di  $n$  nodi rappresentato tramite una struttura dati collegata in cui il record di un nodo  $v$  contiene le seguenti informazioni: il valore del nodo  $k(v)$  e due puntatori  $s(v)$  e  $d(v)$  rispettivamente al figlio sinistro e al figlio destro. La *profondità* di un nodo  $v$  è la distanza, misurata come numero di archi, del cammino dalla radice a  $v$ . Il livello  $i$  di  $T$  è l'insieme di nodi che hanno tutti profondità  $i$ . Si progetti un algoritmo con complessità temporale  $O(n)$  che calcoli se esiste un livello la cui somma dei valori dei nodi è almeno un certo valore target (preso in input)  $\Delta$ . Si fornisca lo pseudocodice dettagliato dell'algoritmo.

**Esercizio 3 [12 punti]**

Sia  $G = (V, E)$  un grafo orientato e non pesato, e siano  $s$  e  $t$  due nodi, e si assuma che  $t$  non è raggiungibile da  $s$  in  $G$ . Avete a disposizione una mossa, una sola. La mossa consiste nel prendere un arco di  $G$  e cambiargli orientamento. Progettare un algoritmo efficiente che trova, se esiste, una mossa dopo la quale  $t$  è diventato raggiungibile da  $s$ .