

Algoritmi e Strutture Dati (modulo I)

Testo della prova scritta del 16 luglio 2018

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: 3^n , $\frac{n^2+n^3 \log n}{n+\sqrt{n+3}}$, $n^2 \sqrt[50]{\log^2 n + 3}$, 2^n , $n^2 \log n^3$, $\frac{n^2+3}{\log \log n}$, $(n^2 + \sqrt{n}) \log n$, 2^{n+1024} , $n^2 + n^{1.5} \log^{3.5} n$.

- (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:

$$T_1(n) = 2T_1(n - 20) + 20, T_1(1) = 1;$$

$$T_2(n) = 1024 T_2(n/2) + n^2, T_2(1) = 1;$$

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 colore del nodo $c(v) \in \{B, N\}$ che può essere *bianco* (B) o *nero* (N), un puntatore $p(v)$ al padre e due puntatori $s(v)$ e $d(v)$ rispettivamente al figlio sinistro e al figlio destro. La profondità di un nodo v è la lunghezza (intesa come numero di archi) del cammino fra la radice e v . Diciamo che un nodo v è *scuro* se il numero di antenati di v di colore nero è strettamente maggiore di quello dei suoi antenati bianchi¹.

Progettare un algoritmo con complessità temporale $O(n)$ che, preso T e un parametro intero h , restituisca il numero di nodi scuri di T che hanno profondità almeno h . Si fornisca lo pseudocodice dettagliato dell'algoritmo, analizzandone la complessità.

Esercizio 3 [12 punti]

Sia $G = (V, E)$ un grafo non orientato e non pesato, dove ogni arco $e \in E$ ha uno fra tre possibili colori $\{1, 2, 3\}$, e sia $U \subseteq V$ un sottoinsieme di nodi *speciali*. Su un nodo $s \in V$ è posizionata una pedina, inizialmente di colore 1. Il vostro obiettivo è quello di portare la pedina su un nodo *target* $t \in V$. Per far ciò potete usare le seguenti *mosse*:

- **sposta**: se la pedina si trova sul nodo $u \in V$, ha colore i , e c'è un arco $(u, v) \in E$ di colore i , allora potete spostare la pedina da u a v . Il colore della pedina resta i ;
- **cambia colore**: se la pedina si trova su un nodo speciale $u \in U$, potete ricolorare la pedina del colore che volete. La pedina in questo caso, dopo la mossa, resta sul nodo u .

Progettate un algoritmo efficiente che trova, se ne esiste una, la sequenza più corta di mosse che risolve l'istanza del gioco.

¹Si ricorda che gli antenati di un nodo v in un albero sono tutti i nodi lungo il cammino da v alla radice, v compreso.