

Elementi di Algoritmi e Strutture Dati

Testo della prova scritta del 2 febbraio 2010

docente: Luciano Gualà

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

Esercizio 1 [8 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: $n \log n$, $n^2 + \log n$, ϕ^n , $\log^4 n$, 1 , $\log \log n$, 2^n , $n^{3/2}$, $2^n \log n$, 2^{50} , $\log n^4$, $n^2 - 12\sqrt{n}$, 2^{2n} , $\sqrt{\log n}$, $n^n - 2^n$, $12 \log n$.

- (b) Siano $f(n), g(n) : \mathbb{N} \rightarrow \mathbb{N}^+$ due funzioni con $g(n) = \Theta(n)$. Dimostrare o confutare la seguente relazione:

$$f(g(n)) = \Theta(f(n)).$$

Esercizio 2 [10 punti] Sia A una matrice $n \times n$ di valori non negativi. Per ogni coppia di indici $1 \leq i, j \leq n$, definiamo $s_{i,j} = \sum_{h=1}^i \sum_{k=1}^j A[h, k]$.

- (a) Si progetti un algoritmo con complessità temporale $O(n^3)$ che, data la matrice A , calcola una matrice B $n \times n$ definita nel seguente modo: per ogni $1 \leq i, j \leq n$, $B[i, j] = s_{i,j}$.
- (b) Si progetti un algoritmo che, data la matrice A , la matrice B calcolata nel punto precedente, e un valore x , verifichi se esistono due indici i, j tale che $s_{i,j} = x$.

Attenzione: l'esercizio sarà valutato solo se corredato da adeguata descrizione del funzionamento dell'algoritmo, in base ai seguenti parametri: correttezza, efficienza e analisi di complessità.

Esercizio 3 [8 punti]

- (a) A partire da un albero AVL vuoto, si mostrino le modifiche apportate all'albero in seguito ai seguenti inserimenti: 5, 10, 15, 20, 25, 30, 12, 16, 11.
- (b) Per un problema sono noti tre algoritmi ricorsivi, A_1 , A_2 e A_3 , le cui complessità temporali sono descritte dalle seguenti equazioni di ricorrenza:

$$T_1(n) = 4T_1(n/2) + \sqrt{n} \log n + n^{9/5}, T_1(1) = 1;$$

$$T_2(n) = T_2(n-1) + n + 1, T_2(1) = 1;$$

$$T_3(n) = 3T_3(n/3) + n \log n, T_3(1) = 1.$$

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

Esercizio 4 [8 punti] Si dia la definizione di Heap, poi si descriva la procedura di `Heapify` che, preso un vettore A di n elementi, trasforma A in un Heap. Si derivi in particolare la complessità temporale di `Heapify`.