

Algoritmi e Strutture Dati (modulo I)
Testo della prova scritta del 28 gennaio 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: $n^3 \log \log n$, $\frac{n^2 \sqrt{n^3+7}}{\sqrt{n+3 \log^3 n}}$, $n^2 \sqrt{n} + n^3$, 2^n , $\frac{n^3}{\log n^{33}}$, $\frac{n^3+313}{\log n}$, 3^{3n} , 3^n , $n^{3.00001}$.

- (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:

$$T_1(n) = T_1(n - 5) + 100, T_1(1) = 1;$$

$$T_2(n) = T_2(99n/100) + n, T_2(1) = 1;$$

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

Esercizio 2 [12 punti] Sia $A[1 : n]$ un vettore di n numeri, e $B[1 : n]$ un vettore di n elementi. Un generico elemento di B è una coppia (v, c) , dove v è un numero che chiameremo il *valore* dell'elemento e $c \in \{R, B\}$ è il *colore* dell'elemento, diciamo *rosso*, o *blu*. Progettare un algoritmo che, dati A, B e un intero k , calcoli quanti elementi di A hanno almeno k elementi rossi in B di valore strettamente più piccolo (del valore dell'elemento di A). L'algoritmo deve avere una complessità temporale $o(n^2)$.

Esercizio 3 [12 punti]

La rete stradale su cui viaggiate è modellata con un grafo orientato e pesato $G = (V, E, w)$, dove il peso $w(u, v)$ di un generico arco $(u, v) \in E$ rappresenta il tempo di percorrenza dell'arco. Voi siete su $s \in V$ e volete arrivare sul nodo $t \in V$ nel minor tempo possibile. La vostra macchina è equipaggiata con un moderno meccanismo di turbo. Se attivato lungo un arco, questo meccanismo permette di dimezzare il tempo di percorrenza dell'arco. Il meccanismo però può essere attivato una volta sola.

- Mostrare che la soluzione migliore non consiste sempre nell'attivare il turbo lungo il cammino minimo in G da s a t .
- Progettate un algoritmo efficiente che calcola una soluzione ottima, ovvero una strategia che vi porta a t nel minor tempo possibile.