Algoritmi e Strutture Dati (modulo I)

Testo della prova scritta del 18 giugno 2014

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-10}, $\frac{n^2+1}{\log\log n}$, $n^2 + \sqrt{n+96}$, 2^n, $n\sqrt{n+9}\log^6 n$, $\frac{(1+n)\sqrt[3]{n^6+\log^6 n}}{n^{0.5}\sqrt{\log n+3n}}$, $\frac{n^2+1}{\log^2\log n}$, $2^{\frac{n}{10}}$, $\frac{n^2+1}{\log\log^2 n}$.

(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:

$T_1(n) = 3T_1(n-1) + 3$;

$T_2(n) = 27\,T_2(n/3) + n^{2.5}\log^{3.5} n + n^3$;

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

Esercizio 2 [12 punti] Siano $A[1:n]$ e $B[1:n]$ due vettori di n numeri positivi. Diciamo che un *accoppiamento* è una coppia di indici (i,j) tale che $A[i] = B[j]$. Progettare un algoritmo con complessità temporale $o(n^2)$ che, dati A e B, calcoli il numero totale di accoppiamenti. Si faccia attenzione al fatto che sia A che B possono contenere elementi ripetuti.

Esercizio 3 [13 punti] *Il signor Marche va a Venezia*

Il signor Marche si è recato in vacanza a Venezia e ha scoperto, con suo disappunto, quanto è cara la famosa città veneta. Ora si trova nella condizione di dover attraversare un canale in gondola e, guardando i prezzi sui volantini, gli è preso un colpo. Questo canale, che può essere percorso in un'unica direzione, passa per n diversi porticcioli, che per comodità numereremo da 1 a n. In ogni porticciolo è possibile imbarcarsi su una gondola e scendere lungo il canale verso porti successivi. Il signor Marche si trova nel porto 1 e deve raggiungere il porto n. Il costo di imbarco dipende dal porto di partenza e da quello di arrivo. Più precisamente, imbarcarsi nel porto i e scendere nel porto j costa $c(i,j)$ $(i > j)$. Progettate un algoritmo di programmazione dinamica che aiuti il signor Marche a raggiungere l'ultimo porto spendendo il meno possibile e appagando così, almeno in parte, il suo proverbiale attaccamento ai soldi. L'algoritmo, che deve calcolare la miglior soluzione prima del tramonto, deve avere complessità polinomiale nella dimensione dell'istanza.