

Algoritmi e Strutture Dati con Laboratorio (modulo I)

Testo della prova scritta del 13 giugno 2013

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^{2n} , $\frac{n^2\sqrt{n^3+1}}{n+1}$, $n^2\sqrt{n} + n \log^9 n$, 2^n , $n^{2.5} \log n^{2.5}$, $n^{2.5}(7 + \log \log n)$, $2^{\sqrt{\log n}}$, $\frac{7 \sqrt[10]{n} - \log n}{3}$, $n^{2.501}$.

- (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:

$$T_1(n) = 2T_1(n-1) + \sqrt{n}, \quad T_1(1) = 1;$$

$$T_2(n) = 16T_2(n/2) + n\sqrt[3]{n \log n} + n^4, \quad T_2(1) = 1;$$

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

Esercizio 2 [12 punti] Sia $V[1:n]$ un vettore di n valori positivi. Progettare un algoritmo che, dato V e un valore x , dica se esistono due indici (anche non distinti) i e j tali che $\sum_{k=1}^i V[k] + \sum_{k=1}^j V[k] = x$. L'algoritmo deve avere complessità temporale $o(n^2)$.

Esercizio 3 [13 punti] Il signor Marche è tornato da un lungo periodo di vacanza fra Roma e Firenze e sta per affrontare un lungo periodo di lavoro che durerà n giorni. Il suo è un lavoro a cottimo e i soldi che riesce a portarsi a casa dipendono da *quanto*, *quando* e *come* lavorerà. Lui sa lavorare in due modi: *forte* e *piano*. Se nel giorno i lavora forte guadagnerà f_i euro, mentre se lavora piano guadagnerà p_i euro. Chiaramente $f_i \geq p_i$. A rendere le cose difficili ci sono dei vincoli di stanchezza. Dopo aver lavorato per un giorno in modo forte, il signor Marche ha bisogno di riposarsi (e quindi di non lavorare) per i successivi due giorni. Inoltre, in ogni caso lui non riesce a lavorare due giorni di seguito. Come potete immaginare i giorni in cui non lavora non guadagna niente.

Progettare un algoritmo di programmazione dinamica che calcoli un piano di lavoro per il signor Marche che gli faccia guadagnare il più possibile.