

Algoritmi e Strutture Dati (modulo I)
Testo della prova scritta del 18 febbraio 2016
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} , $\frac{n^3\sqrt{n+1}}{\sqrt{n^4+n(n^2+1)}}$, $n\sqrt{n+6} + n \log n$, 2^n , $n^2 \log \log n$, $n^2\sqrt{\log n}$, $\frac{n^3+1}{n^{0.9}-1}$, $2^{n \log n}$, $\sqrt{1+n^4 \log \log n}$.

- (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:

$$T_1(n) = 2T_1(n-4) + 1; T_1(1) = 1;$$

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

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

Esercizio 2 [12 punti] Sia $V[1 : n]$ un vettore di n elementi in cui ogni elemento è di colore bianco o nero, ovvero, $V[i] \in \{B, N\}$, per ogni $i = 1, \dots, n$. Dato un intero non negativo $h \leq n-1$, diciamo che un elemento in posizione i è *h-nero-circondato* se l'elemento i ha almeno un elemento nero fra gli h elementi subito alla sua sinistra, e ha almeno un elemento nero fra gli h elementi subito alla sua destra; più formalmente, l'elemento i è *h-nero-circondato* se entrambi i sottovettori $V[\max\{1, i-h\} : i]$ e $V[i, \min\{n, i+h\}]$ contengono almeno un elemento nero. Si noti che un elemento nero è sempre *h-nero-circondato* per ogni valore di h . Progettare un algoritmo che, preso il vettore e il valore h , restituisce il numero di elementi che sono *h-nero-circondati*. L'algoritmo deve avere complessità $O(n)$ anche quando $h = \omega(1)$.

Esercizio 3 [13 punti]

Un drone equipaggiato con una batteria a energia solare si muove su un territorio descritto da una matrice M con n righe e m colonne. Deve andare dal punto in alto a sinistra, di coordinate $(1, 1)$, al punto in basso a destra, di coordinate (n, m) . Soffia un forte vento che impedisce al drone di spostarsi in tutte le direzioni. In particolare, il drone può spostarsi ogni volta solo di una cella a destra o di una cella in basso rispetto alla posizione in cui si trova, operazione che gli fa consumare una unità di energia della batteria. La batteria ha un capacità massima di Δ unità, all'inizio completamente carica. Ovviamente il drone non può muoversi se il livello della sua batteria è a 0. Esso, però, è equipaggiato di pannelli ricettivi che gli permettono di ricaricare la batteria qualora passasse in punti in cui i raggi solari sono particolarmente forti. Più precisamente, per ogni posizione (i, j) , è noto il numero $M[i, j] \geq 0$ di unità di energia che si ottengono quando si passa nella posizione (i, j) . La batteria non può accumulare più di Δ unità di energia, ma non si danneggia se sottoposta a raggi solari quando è già al massimo della carica. Progettare un algoritmo di programmazione dinamica che calcoli quale è il massimo livello di batteria con cui il drone può arrivare a destinazione.