

Algoritmi e Strutture Dati (modulo I)

Testo della prova scritta del 20 luglio 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: $(\frac{5}{2})^n$, $\frac{n^2 \sqrt[3]{n^6+8}}{n^3+\sqrt{n+8}}$, $\sqrt[50]{n^{51} + 51}$, 2^n , $\sqrt{n} + n \log n^4$, $(n+4) \log \log n$, $n + \sqrt{n} \log^{22} n$, $2^{\log^2 n}$, $n + n^{1.5} \log^{1.5} n$.

- (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:

$$T_1(n) = T_1(\frac{2}{3}n) + n, T_1(1) = 1;$$

$$T_2(n) = 2T_2(n-2) + 2, T_2(1) = 1;$$

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

Esercizio 2 [12 punti] Sia $A[1 : n]$ e $B[1 : n]$ due vettori di n numeri positivi ciascuno. Progettare un algoritmo che, dato un valore x , trova la coppia di indici i e j tale che la somma $A[i] + B[j]$ non supera x ed è il più vicino possibile in valore a x , ovvero la coppia di indici i, j tale che (i) $A[i] + B[j] \leq x$, e (ii) per ogni altra coppia di indici i', j' tale che $A[i'] + B[j'] \leq x$ vale $A[i'] + B[j'] \leq A[i] + B[j]$. L'algoritmo deve avere complessità $o(n^2)$.

Esercizio 3 [13 punti]

Nella versione Beta di Pokemon Go la mappa è una sola strada di $n + 1$ tratte. Tu sei nella posizione 0 e ad ogni istante di tempo sei costretto a spostarti a destra di una posizione, dalla posizione corrente i alla posizione $i + 1$. In ogni posizione $i \geq 1$ ti si presenta davanti un pokemon. Tu puoi fare una sola delle seguenti due mosse:

- raccogliere le k_i pokemon ball che sono nella posizione i ;
- lanciare al pokemon delle pokemon ball al fine di catturarlo. Il numero di ball che ti permettono di catturare il pokemon nella posizione i è p_i e tu, chiaramente, ne devi precedentemente aver raccolte a sufficienza.

All'inizio, quando sei nella posizione 0, non possiedi pokemon ball. Chiaramente $\Delta = \sum_{i=1}^n k_i$ è il numero massimo di pokemon ball che puoi raccogliere nell'intero percorso. Progettare un algoritmo di programmazione dinamica che calcoli il numero massimo di pokemon catturabili. L'algoritmo deve avere complessità polinomiale in n e Δ .