Algoritmi e Strutture Dati (modulo I)

Testo della prova scritta del 16 giugno 2015

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: $2^{n+\sqrt{n}}$, $\frac{n^3 \log n}{\sqrt{n^4+n(n^2+1)}}$, $\frac{n}{4} + n \log n$, 2^n, $n \log\log n$, $n\sqrt{\log n}$, $\frac{n^2+1}{n^{0.99}-1}$, $2^{n \log n}$, $\sqrt{1+n^2 \log\log n}$.

(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:

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

$T_2(n) = T_2(n/3) + n;$

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

Esercizio 2 [12 punti] Sia T un albero binario di n nodi rappresentato tramite una struttura dati collegata in cui il record di un nodo v contiene le seguenti informazioni: la chiave del nodo $c(v)$, un puntatore $p(v)$ al padre e due puntatori $s(v)$ e $d(v)$ rispettivamente al figlio sinistro e al figlio destro. La profondità di un nodo v è la lunghezza (intesa come numero di archi) del cammino fra la radice e v. Progettare un algoritmo con complessità temporale $O(n)$ che, preso T e un parametro intero h, restituisca il puntatore ad un nodo di chiave minima fra tutti i nodi con profondità almeno h. Si fornisca lo pseudocodice dettagliato dell'algoritmo.

E se si volessero calcolare, fra i nodi di profondità almeno h, i k nodi di chiave più piccola, invece che semplicemente il minimo?

Esercizio 3 [13 punti]

Sia S una sequenza di n interi, $s_1, \ldots, s_n$, e sia ℓ un intero. Si vuole dividere S in sottosequenze contigue tale che la somma di ogni sottosequenza è il più possibile vicino ad ℓ. La *penalità* relativa ad una sottosequenza di elementi $S[i,j] = s_i, s_{i+1}, \ldots, s_j$ è definita come $cost(i,j) = |\ell - \sum_{k=i}^{j} s_k|^2$. Si vuole dividere S in modo tale che la somma delle penalità sia minima. Più formalmente, si vogliono trovare degli indici $1 \leq i_1 < i_2 \cdots < i_h \leq n$ tale che la seguente *penalità complessiva* è minimizzata:

$$cost(1, i_1) + cost(i_1 + 1, i_2) + cost(i_2 + 1, i_3) \cdots + cost(i_h, n).$$

A titolo di esempio si consideri la sequenza $S = 5, 4, 3, 4, 7, 10$. Per $\ell = 10$, la soluzione migliore è dividere S nelle tre sottosequenze $5, 4, 3 \mid 4, 7 \mid 10$; la penalità complessiva di tale soluzione è pari a $2^2 + 1^2 + 0^2 = 5$.

Progettare un algoritmo di programmazione dinamica che calcoli la penalità complessiva minima ottenibile da S.