Algoritmi e Strutture Dati (modulo I) - testo prova scritta 21/01/2025

docenti: Luciano Gualà

Cognome:..... Nome:..... Matr.:.....

Esercizio 1 [16 punti]

A: notazione asintotica. Dire quali delle seguenti relazioni asintotiche sono vere:

$n + n\log^2 n = o(n^{0.99}\log^{30} n)$; $\sqrt[3]{\log n} = o(\log\log^{30} n)$; $n^2 = \Omega(\frac{n^{2.001}}{\log^{2001} n})$; $\frac{n\sqrt{n}+\log n}{\sqrt{n^3+3}} = \Theta(\sqrt{n})$;

$4^n = \omega(3^n)$; $2^n = \Theta(2^n + n^3)$; $2^{n/2} = o(2^n + n^2)$; $2^n = \Theta(2^{n+2^4})$;

B: equazioni di ricorrenza. Fornire la soluzione asintotica alle seguenti relazioni di ricorrenza:

$T(n) = 2T(n/2) + n$; Soluzione:

$T(n) = T(n-1) + n^2$; Soluzione:

C: algoritmi e complessità. Quale algoritmo useresti e quanto costa se devi:

- Cercare un elemento in una lista ordinata implementata con record e puntatori:
- Ordinare n interi compresi fra 1 e n^2:
- Capire in un grafo diretto e pesato quanto è lungo il cammino più corto da s a t che non passa per un certo vertice u:
- In un grafo non orientato e pesato con pesi tutti maggiori di 100, calcolare la distanza da s a t:

Esercizio 2 [8 punti]

Ti è dato in input un array $A[1:n]$ di n numeri, con n pari. L'obiettivo è partizionare gli n numeri in $n/2$ gruppi da 2 numeri in modo che la somma dei numeri all'interno di ogni gruppo sia uguale per tutti i gruppi.

Progetta un algoritmo che, dato A restituisce `true` se questo è possibile, `false` altrimenti. Il punteggio pieno è pensato per soluzioni di complessità temporale $o(n^2)$, ma anche soluzioni con complessità $\Theta(n^2)$ saranno prese in considerazione.

Esercizio 3 [8 punti]

Sia $G = (V, E)$ un grafo orientato con n nodi ed m archi. Ad ogni arco $e \in E$, è associato un costo non negativo di attraversamento $w(e)$, e un colore $col(e) \in \{c_1, c_2\}$. Inoltre, ogni nodo $v \in V$ ha associato due costi, $w_1(v)$ e $w_2(v)$. Intuitivamente, $w_i(v)$ è il costo per attraversare il nodo v se si arriva a v con un arco di colore c_i.

Dati due nodi s a t e un cammino π da s a t, il costo di attraversamento di π è definito come la somma dei costi di attraversamento degli archi di π più la somma dei costi di attraversamento dei nodi di π.

Si progetti un algoritmo che, dati due nodi s e t, calcoli il cammino da s a t di costo di attraversamento minimo in G. L'algoritmo deve avere complessità temporale $O(m + n\log n)$.