Università degli Studi dell'Aquila

Prova Scritta di **Algoritmi e Strutture Dati con Laboratorio**

Martedì 4 Luglio 2017 – Prof. Guido Proietti (Modulo di Teoria)

Scrivi i tuoi dati ⟹	Cognome:	Nome:	Matricola:	PUNTI
ESERCIZIO 1	Risposte Esatte:	Risposte Omesse:	Risposte Errate:	

ESERCIZIO 1: Domande a risposta multipla

Premessa: Questa parte è costituita da 10 domande a risposta multipla. Per ciascuna domanda vengono fornite 4 risposte, di cui soltanto una è corretta. Per rispondere utilizzare la griglia annessa, barrando con una $\times$ la casella corrispondente alla risposta prescelta. È consentito omettere la risposta. In caso di errore, contornare con un cerchietto la $\times$ erroneamente apposta (ovvero, in questo modo $\otimes$) e rifare la $\times$ sulla nuova risposta prescelta. Se una domanda presenta più di una risposta, verrà considerata omessa. Per tutti i quesiti verrà attribuito un identico punteggio, e cioè: risposta esatta 3 punti, risposta omessa 0 punti, risposta sbagliata -1 punto. Il voto relativo a questa parte è ottenuto sommando i punti ottenuti e normalizzando su base 30. Se tale somma è negativa, verrà assegnato 0.

1. $f(n) = \Omega(g(n))$ se:
   a) Esistono due costanti $c \geq 0$ e $n_0 \geq 0$ tali che $f(n) \geq c \cdot g(n)$ per ogni $n \geq n_0$.
   b) Esistono due costanti $c > 0$ e $n_0 \geq 0$ tali che $f(n) \leq c \cdot g(n)$ per ogni $n \geq n_0$.
   c) Esistono due costanti $c > 0$ e $n_0 \geq 0$ tali che $f(n) \geq c \cdot g(n)$ per ogni $n \leq n_0$.
   *d) Esistono due costanti $c > 0$ e $n_0 \geq 0$ tali che $f(n) \geq c \cdot g(n)$ per ogni $n \geq n_0$.
2. Un problema ha una delimitazione inferiore alla complessità temporale $\Omega(f(n))$ se:
   *a) Tutti gli algoritmi per la sua risoluzione hanno una delimitazione inferiore alla complessità computazionale pari a $\Omega(f(n))$
   b) Tutti gli algoritmi per la sua risoluzione hanno una delimitazione superiore alla complessità computazionale pari a $O(f(n))$
   c) Esiste un algoritmo per la sua risoluzione che ha una delimitazione inferiore alla complessità computazionale pari a $\Omega(f(n))$
   d) Esiste un algoritmo per la sua risoluzione che ha una delimitazione superiore alla complessità computazionale pari a $O(f(n))$
3. L'altezza dell'albero di decisione associato al problema dell'ordinamento di n elementi basato su confronti è:
   *a) $\Omega(n \log n)$ b) $\omega(n \log n)$ c) $O(n \log n)$ d) $\Theta(n!)$
4. A quale delle seguenti classi di complessità <u>non</u> appartiene la complessità dell'algoritmo Quicksort:
   a) $O(n^3)$ b) $\Theta(n^2)$ c) $\Omega(n)$ *d) $o(n^2)$
5. La procedura *Heapify* per la costruzione di un heap binario applicata ad $A = [5, 7, 6, 8, 9]$ restituisce:
   a) $A = [9, 8, 7, 5, 6]$ *b) $A = [9, 8, 6, 5, 7]$ c) $A = [9, 7, 8, 6, 5]$ d) $A = [9, 8, 5, 6, 7]$
6. Sia dato un albero AVL di altezza h, e si supponga di inserire in esso un elemento che provochi lo sbilanciamento dell'albero. Quale tra le seguenti è una coppia di valori ammissibili per l'altezza dell'AVL prima e dopo aver effettuato il ribilanciamento complessivo dell'AVL?
   a) $h, h+1$ b) $h, h-1$ c) $h-1, h$ *d) h, h
7. Si consideri il grafo $G = (V, E)$ con $V = \{1, 2, 3, 4, 5\}$ ed $E = \{(1, 3), (1, 4), (2, 3), (2, 4), (3, 5)\}$. Quali delle seguenti affermazioni è <u>vera</u>:
   *a) G è bipartito b) G è disconnesso c) G è aciclico d) G è bipartito completo
8. Si consideri il grafo di cui alla domanda (7), e si pesino gli archi con peso ciascuno pari alla somma dei numeri associati ai due vertici estremi. Qual è la sequenza di vertici aggiunti alla soluzione dall'algoritmo di Dijkstra per determinare l'albero dei cammini minimi radicato nel vertice 1?
   a) $\langle 1, 2, 3, 4, 5 \rangle$ b) $\langle 1, 3, 2, 4, 5 \rangle$ *c) $\langle 1, 3, 4, 2, 5 \rangle$ d) $\langle 1, 4, 2, 3, 5 \rangle$
9. Sia dato un grafo pesato $G = (V, E)$ con n nodi ed m archi, senza cicli negativi, e si consideri il problema di trovare i cammini minimi in G tra tutte le coppie di nodi. Quando è conveniente (asintoticamente) applicare l'algoritmo di Floyd&Warshall rispetto ad un'applicazione ripetuta dell'algoritmo di Dijkstra con heap binari?
   *a) $m = \omega(n^2/\log n)$ b) $m = \Theta(n)$ c) per ogni valore di m d) per nessun valore di m
10. Dato un grafo connesso di n nodi ed m archi, per quale valore (asintotico) di m si ha che l'implementazione di Prim con array non ordinati ha <u>la stessa</u> complessità temporale dell'algoritmo di Borůvka?
    *a) $m = \Theta(n^2/\log n)$ b) $m = \Theta(n^2)$ c) per ogni valore di m d) per nessun valore di m

Griglia Risposte

	Domanda									
Risposta	1	2	3	4	5	6	7	8	9	10
a										
b										
c										
d										