Università degli Studi dell'Aquila

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

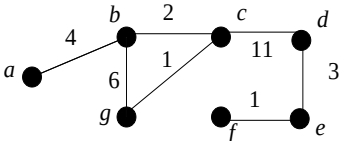
Martedì 3 Febbraio 2015 – Prof. Guido Proietti (Modulo di Teoria)

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

ESERCIZIO 1: Domande a risposta multipla

Premessa: La prova 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 finale è dato dalla somma dei punti ottenuti. Se tale somma è negativa, verrà assegnato 0.

1. L'algoritmo di ordinamento non crescente Insertion Sort applicato ad $A = [1, 2, 4, 5, 3]$, esegue un numero di confronti tra elementi pari a:
   a) 5 *b) 9 c) 10 d) 4
2. Quale delle seguenti relazioni asintotiche è vera:
   a) $n \log n^2 = o(n \log n^3)$ b) $n \log n^2 = \omega(n \log n)$ c) $n \log^2 n = \Theta(n \log n)$ *d) $n \log n^2 = \Omega(n \log n)$
3. A quale delle seguenti classi <u>non</u> appartiene la complessità dell'algoritmo Merge Sort:
   *a) $\omega(n \log n)$ b) $\omega(n)$ c) $\Omega(n \log n)$ d) $o(n^2)$
4. La procedura deleteMax(A) applicata alla coda di priorità rappresentata tramite vettore non ordinato $A = [12, 9, 3, 6, 5, 2]$, restituisce:
   a) $A = [9, 3, 6, 5, 2, nil]$ b) $A = [12, 9, 3, 6, 5, 2]$ c) $A = [2, 9, 3, 6, 5]$ *d) $A = [2, 9, 3, 6, 5, nil]$
5. Dato un albero AVL T di altezza $\Theta(\log n)$, si consideri la cancellazione di una sequenza di $\Theta(\log n)$ elementi da T. L'altezza dell'AVL risultante sarà:
   a) $O(1)$ b) $o(\log n)$ c) $\log n$ *d) $\Theta(\log n)$



6. Si consideri il grafo e si costruisca l'albero dei cammini minimi radicato in a. Quale tra i seguenti rappresenta il vettore delle distanze finali da a (le distanze appaiono secondo l'ordine alfabetico dei nodi):
   a) {0,4,6,7,17,20,21} *b) {0,4,6,17,20,21,7} c) {0,4,2,11,3,11,1} d) {0,1,2,3,4,5,2}
7. Quale tra i seguenti rappresenta lo pseudocodice dell'algoritmo di Bellman&Ford:

   a)
   B&F($G = (V, A, w), s \in V$)
   $D_{sv} = +\infty$ per $v \neq s, D_{ss} = 0$
   for $i = 1$ to $n - 1$ do
   for each $(u, v) \in A$ do
   if $D_{su} + w(uv) > D_{sv}$ then
   $D_{sv} = D_{su} + w(uv)$
   return D

   b)
   B&F($G = (V, A, w), s \in V$)
   $D_{sv} = +\infty$ per $v \neq s, D_{ss} = 0$
   for $i = 1$ to $n - 1$ do
   for each $(u, v) \in A$ do
   if $D_{su} + w(uv) < D_{sv}$ then
   $D_{su} = D_{sv} + w(uv)$
   return D

   *c)
   B&F($G = (V, A, w), s \in V$)
   $D_{sv} = +\infty$ per $v \neq s, D_{ss} = 0$
   for $i = 1$ to $n - 1$ do
   for each $(u, v) \in A$ do
   if $D_{su} + w(uv) < D_{sv}$ then
   $D_{sv} = D_{su} + w(uv)$
   return D

   d)
   B&F($G = (V, A, w), s \in V$)
   $D_{sv} = +\infty$ per $v \neq s, D_{ss} = 0$
   for $i = 1$ to $n - 1$ do
   for each $(u, v) \in A$ do
   if $D_{su} + w(uv) = D_{sv}$ then
   $D_{sv} = D_{su} + w(uv)$
   return D

8. Si consideri il grafo di domanda 6, si enumerino i vertici in ordine alfabetico, da $v_1 = a$ fino a $v_7 = g$, e si orientino gli archi da vertice con numero inferiore a vertice con numero superiore. Quanto è lungo il cammino minimo 2-vincolato tra a e g?
   *a) 10 b) 2 c) $+\infty$ d) 7
9. Si consideri il grafo di domanda 6, e si applichi l'algoritmo di Prim con nodo sorgente c. Qual è la sequenza di vertici aggiunti alla soluzione?
   a) $\{c, b, g, a, d, e, f\}$ *b) $\{c, g, b, a, d, e, f\}$ c) $\{c, g, b, d, a, e, f\}$ d) $\{c, d, e, f, g, b, a\}$
10. Dato un grafo connesso di n nodi ed m archi, per quale valore (asintotico) di m si ha che l'implementazione di Kruskal con alberi QuickFind bilanciati ha la stessa complessità temporale dell'algoritmo di Borůvka?
    *a) sempre b) solo per $m = \Theta(n^2)$ c) solo per $m = \Theta(n \log n)$ d) mai

Griglia Risposte

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