

Università degli Studi dell'Aquila

Seconda Prova Parziale di **Algoritmi e Strutture Dati con Laboratorio**

Mercoledì 27 Gennaio 2016 – Prof. Guido Proietti (Modulo di Teoria)

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

ESERCIZIO 1: Domande a risposta multipla

Premessa: Per ciascuna domanda vengono fornite 4 risposte, di cui soltanto una è corretta. Per rispondere utilizzare la griglia annessa, barrando con una x la casella corrispondente alla risposta prescelta. È consentito omettere la risposta. In caso di errore, contornare con un cerchietto la × erroneamente apposta (ovvero, in questo modo ⊗) e rifare la × 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 ottenuto sommando i punti ottenuti e normalizzando su base 30. Se tale somma è negativa, verrà assegnato 0.

- 1. In un albero AVL di n elementi, la cancellazione della radice comporta nel caso migliore un numero di rotazioni pari a: c) $\Theta(\log n)$ d) 1
- 2. Sia dato un AVL di n elementi nel quale si eseguono in successione O(1) cancellazioni e $\Theta(\log n)$ inserimenti. Nel caso peggiore, quante rotazioni subirà l'AVL?

a) $\Theta(1)$ b) $\Theta(n)$ c) $\Theta(\log^2 n)$ *d) $\Theta(\log n)$

3. Si supponga di inserire la sequenza di chiavi 26,14,6 (in quest'ordine) in una tavola hash di lunghezza m=3 (ovvero con indici 0, 1, 2) utilizzando l'indirizzamento aperto con funzione hash $h(k) = k \mod 3$, e risolvendo le collisioni con il metodo della scansione quadratica con $c_1 = c_2 = 1$. Quale sarà la tavola hash finale?

a) A = [14, 6, 26]b) A = [26, 6, 14]c) A = [6, 26, 14]*d) A = [6, 14, 26]

- 4. In un grafo con n vertici ed $m = \Theta(n^2)$ archi rappresentato con liste di adiacenza, la verifica di completezza costa: *a) $\Theta(n^2)$ b) $\omega(n^2)$ c) $\Theta(n)$ d) $o(n^2)$
- 5. Si consideri il grafo G = (V, E) con $V = \{1, 2, 3, 4, 5\}$ ed $E = \{(1, 2), (1, 5), (2, 4), (2, 5), (3, 5)\}$. Quali delle seguenti affermazioni è falsa:
 - b) il BFS di G radicato in 2 ha altezza 2 *c) il diametro di G, ovvero la distanza massima tra *a) G è bipartito due nodi in G, è pari a 2 d) G ha grado 3
- 6. Sia dato un grafo connesso G con n vertici, numerati da 1 ad n, ed $m = \Theta(n \log n)$ archi orientati, disposti in modo arbitrario, ma in modo tale da garantire l'aciclicità. Si applichi ora l'algoritmo di ordinamento topologico rispetto al nodo sorgente etichettato 1. La complessità risultante è pari a:

a) $\Theta(n^2)$ b) $\Theta(n)$ *c) $\Theta(n \log n)$ d) indefinita (non è detto che l'algoritmo possa essere applicato)

7. Quale tra i seguenti rappresenta lo pseudocodice dell'algoritmo di Bellman&Ford: *a) $B\&F(G = (V, A, w), s \in V)$ $D_{sv} = +\infty \text{ per } v \neq s, D_{ss} = 0$ for i = 1 to n - 1 do for each $(u, v) \in A$ do if $D_{su} + w(u, v) < D_{sv}$ then $D_{sv} = D_{su} + w(u, v)$

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

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

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

8. 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 md) per nessun valore di m

9. Sia dato un grafo pesato G = (V, E) con n nodi ed m archi, e si consideri il problema di trovare il minimo albero ricoprente di G. Quando è equivalente (asintoticamente) applicare l'algoritmo di Kruskal con alberi QuickFind senza euristica di bilanciamento e l'algoritmo di Prim con heap binari?

a) $m = o(n^2)$ *b) $m = \Omega(n^2/\log n)$ c) per ogni valore di md) per nessun valore di m

10. Sia dato un grafo pesato G = (V, E) con n nodi ed m archi, e si consideri il problema di trovare il minimo albero ricoprente di G. Quando è strettamente conveniente (asintoticamente) l'implementazione di Prim con heap di Fibonacci rispetto all'algoritmo di Borůvka?

b) $m = \Theta(n)$ c) per ogni valore di m*a) $m = \omega(n)$ d) per nessun valore di m

Griglia Risposte

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