Università degli Studi dell'Aquila

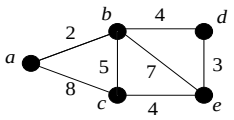
Seconda Prova Parziale di **Algoritmi e Strutture Dati con Laboratorio**

Giovedì 11 Febbraio 2016 – 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: 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 ottenuto sommando i punti ottenuti e normalizzando su base 30. Se tale somma è negativa, verrà assegnato 0.

1. Dato un albero AVL di n elementi, si supponga di dover cancellare un elemento a distanza 4 dalla radice. Nel caso peggiore, quante rotazioni si avranno?
   a) 1 b) 2 c) 3 *d) $\Theta(\log n)$
2. Sia dato un AVL di n elementi nel quale si eseguono in successione $O(\log n)$ cancellazioni e $\Theta(1)$ inserimenti. Nel caso peggiore, quante rotazioni subirà l'AVL?
   a) $\Theta(1)$ b) $\Theta(n)$ c) $\Theta(\log n)$ *d) $\Theta(\log^2 n)$
3. Sia n_h il numero di nodi di T_h, ovvero l'albero di Fibonacci di ordine h, e sia F_h l'h-esimo numero della sequenza di Fibonacci. Risulta:
   a) $n_h = F_{h+1} + 3$ b) $n_h = F_{h+2} - 1$ c) $n_h = F_{h+2} + 1$ *d) $n_h = F_{h+3} - 1$
4. Si supponga di inserire la sequenza di chiavi $29, 17, 9$ (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) $A = [17, 9, 29]$ b) $A = [29, 9, 17]$ c) $A = [9, 29, 17]$ *d) $A = [9, 17, 29]$
5. La visita in ampiezza del grafo  eseguita partendo dal nodo d genera un albero BFS di altezza pari a:
   a) 1 *b) 2 c) 3 d) 4
6. L'albero dei cammini minimi con sorgente a del grafo della domanda 5 ha peso totale:
   a) 18 *b) 14 c) 15 d) 21
7. Sia $G = (V, E)$ un grafo *completo* di 6 vertici, in cui i vertici sono numerati da 1 a 6, ed il peso dell'arco (i, j) è pari al minimo tra i e j. Qual è la distanza tra il vertice 3 e il vertice 6 in G?
   a) 1 *b) 2 c) 3 d) 6
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 strettamente più efficiente (asintoticamente) applicare l'algoritmo di Floyd&Warshall rispetto ad un'applicazione ripetuta dell'algoritmo di Dijkstra con heap di Fibonacci?
   a) $m = \omega(n^2/\log n)$ b) $m = \Theta(n)$ c) per ogni valore di m *d) per nessun valore di m
9. Usando gli alberi *QuickUnion* e l'euristica dell'unione pesata *by size*, il problema della gestione di n insiemi disgiunti sottoposti ad $n - 1$ *Union* ed $m = n^2$ *Find* può essere risolto in:
   a) $\Theta(n)$ b) $\Theta(n + m)$ c) $\Theta(n^2)$ *d) $O(n^2 \log n)$
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 heap di Fibonacci è strettamente più efficiente dell'implementazione di Kruskal con alberi *QuickUnion* con euristica di bilanciamento *union by size*?
    a) $m = \Theta(n)$ b) sempre c) mai *d) $m = \omega(n)$

Griglia Risposte

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