

Università degli Studi di L'Aquila

Seconda Prova Parziale di Algoritmi e Strutture Dati con Laboratorio

Lunedì 4 Febbraio 2013 – 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: 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 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 \otimes) e rifare la x 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.	Si supponga di inserire la sequenza di chiavi $10, 22, 31$ (in quest'ordine) in una tavola hash di lunghezza $m=3$ (ovvero	cor
	indici $0, 1, 2$) utilizzando l'indirizzamento aperto con funzione hash $h(k) = k \mod 3$, e risolvendo le collisioni con il metodo de	ella
	scansione lineare. Quale sarà la tavola hash finale?	

- a) A = [10, 22, 31]b) A = [22, 31, 10]c) A = [31, 22, 10]d) A = [31, 10, 22]
- 2. Si consideri il grafo G = (V, E) con $V = \{1, 2, 3, 4\}$ ed $E = \{(1, 3), (1, 4), (2, 3), (2, 4)\}$. Quali delle seguenti affermazioni è <u>falsa</u>: b) G è connesso c) G è aciclico d) G è bipartito completo a) G è bipartito
- 3. Utilizzando l'algoritmo basato sull'ordinamento topologico, quanto costa calcolare tutte le distanze da un nodo sorgente in un albero di n nodi (si selezioni l'opzione che meglio descrive la complessità temporale dell'algoritmo in termini asintotici)? b) $\Theta(n^2)$ a) *O*(*n*) c) $O(n \log n)$ d) $\Theta(n)$
- 4. Si consideri il grafo di Domanda 2, e si pesino gli archi come segue (si osservi che gli archi non sono orientati): w(1,3) = 1, w(1,4) = 15, w(2,3) = 1, w(2,4) = 2. Applicando l'algoritmo di Dijkstra con node sorgente 1, qual è la sequenza di inserimento dei nodi nella soluzione finale?
 - a) 1, 3, 2, 4 b) 1, 3, 4, 2 c) 1, 2, 3, 4 d) 1, 4, 3, 2
- 5. Dato un grafo connesso di n nodi ed m archi, per quale valore (asintotico) di m si ha che l'implementazione dell'algoritmo di Dijkstra con heap di Fibonacci ha la stessa complessità temporale dell'implementazione con heap binomiali?
 - a) $m = \Theta(n)$ b) $m = \Theta(n^2)$ c) $m = \Theta(n \log n)$ d) mai
- 6. Dato un grafo connesso di n nodi ed m archi, per quale valore (asintotico) di m si ha che l'implementazione ripetuta dell'algoritmo di Dijkstra con heap di Fibonacci ha la stessa complessità temporale dell'implementazione di Floyd e Warshall per trovare le distanze tra tutte le coppie di nodi del grafo (entrami implementati utilizzando liste di adiacenza)? a) $m = \Theta(n)$ b) $m = \Theta(n^2)$ c) $m = \Theta(n \log n)$
- 7. Nel problema della gestione di insiemi disgiunti, quale tra le diverse implementazioni proposte garantisce di poter eseguire nel caso peggiore la Union in $O(\log n)$ e la Find in O(1)?
 - b) QuickFind con union by size c) QuickUnion con union by size d) QuickUnion
- 8. Dato un grafo connesso di n nodi ed m archi, per quale valore (asintotico) di m si ha che le implementazioni di Kruskal con alberi QuickFind ed alberi QuickUnion (senza euristiche di bilanciamento) hanno la stessa complessità temporale?

b) $m = \Theta(n^2)$ c) $m = \Theta(n \log n)$ d) mai

- 9. 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) $m = \omega(n)$
- 10. Dato un grafo connesso con n vertici ed m archi, il numero minimo di archi che l'algoritmo di Borůvka aggiunge alla soluzione alla fine della prima passata è pari a:

d) [n/2]a) n - 1b) $\log n$ c) 1

Griglia Risposte

	Domanda									
Risposta	1	2	3	4	5	6	7	8	9	10
a										
b							- 1	, and	7	
С										
d							100			