Università degli Studi dell'Aquila

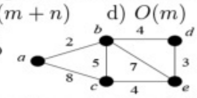
Seconda prova parziale di Teoria del corso di **Algoritmi e Strutture Dati con Laboratorio**

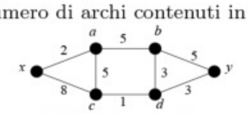
Mercoledì 2 Febbraio 2011 – Prof. Guido Proietti

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

ESERCIZIO 1 (25 punti): Domande a risposta multipla

Premessa: Questa parte è costituita da 20 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 25. Se tale somma è negativa, verrà assegnato 0.

1. Un albero binario di ricerca di altezza k contiene:
   a) meno di $2^{k+1}-1$ elementi b) tra $k+1$ e $2^{k+1}-1$ elementi c) almeno 2^k elementi d) tra 2^k e $2^{k+1}-1$ elementi
2. In un albero AVL di n elementi, la cancellazione di un elemento nel caso migliore induce un numero di rotazioni pari a:
   a) 0 b) 2 c) $\Theta(\log n)$ d) 1
3. Dato un albero AVL T contenente n elementi, si consideri l'inserimento di una sequenza di k elementi in T. La nuova altezza di T diventa: a) $\Theta(n+k)$ b) $\Theta(k+\log n)$ c) $\Theta(\log(n+k))$ d) $\Theta(\log n)$
4. In una tavola ad accesso diretto di dimensione m con un fattore di carico $\alpha = 1\%$, l'inserimento di un elemento di un dizionario di n elementi costa: a) $\Theta(m)$ b) $\Omega(n)$ c) $\Theta(\log n)$ d) $\Theta(1)$
5. Siano $h_1(\cdot), h_2(\cdot)$ due funzioni hash. Quale delle seguenti funzioni descrive il metodo di scansione con hashing doppio in una tabella hash di dimensione m per l'inserimento di un elemento con chiave k dopo l'i-esima collisione:
   a) $c(k,i) = (h_1(k) + m \cdot h_2(k)) \mod i$ b) $c(k,i) = (h_1(k) + h_2(k)) \mod m$
   c) $c(k,i) = (h_1(k) + i \cdot h_2(k)) \mod m$ d) $c(k,i) = (h_1(k) + h_2(k)) \mod i$
6. Siano X e Y due stringhe di lunghezza m ed n. Qual è la complessità dell'algoritmo per la determinazione della distanza tra X e Y basato sulla tecnica della programmazione dinamica?
   a) $O(mn)$ b) $O(n)$ c) $O(m+n)$ d) $O(m)$
7. La visita in profondità del grafo 

   eseguita partendo dal nodo d produce un albero DFS di altezza massima:
   a) 1 b) 2 c) 4 d) 5
8. Un grafo $G=(V,E)$ si dice *bipartito* se l'insieme V può essere partizionato in due sottoinsiemi V_1, V_2 tali che tutti gli archi in E hanno un nodo in V_1 e l'altro in V_2. Sia dunque $G = (V_1 \cup V_2, E)$ un grafo bipartito tale che $|V_1| = 4, |V_2| = 3$. Quanti archi sono necessari affinché G sia connesso? a) 3 b) 4 c) 5 d) 6
9. In un grafo *non completo* con 6 vertici, il massimo numero di archi contenuti in un suo sottografo indotto da 5 vertici qualsiasi è:
   a) 15 b) 5 c) 10 d) 4
10. Qual è la *distanza* tra x ed y nel grafo non orientato: 

    a) 12 b) 11 c) 3 d) 4
11. Si orientino gli archi verticali del grafo della domanda 10 dall'alto verso il basso, e i rimanenti archi da sinistra verso destra. Quali tra i seguenti è un ordinamento topologico dei veritici del grafo:
    a) $< x, a, c, b, d, y >$ b) $< x, c, a, b, d, y >$ c) $< x, a, c, d, b, y >$ d) $< y, d, b, c, a, x >$
12. L'algoritmo di Bellman e Ford applicato ad un grafo pesato con un numero di archi $m = \Theta(n)$, ha complessità:
    a) $\Theta(n^2)$ b) $\Theta(n+m)$ c) $\Theta(n^3)$ d) $O(m \log n)$
13. Dato un grafo pesato e completo con n vertici rappresentato tramite liste di adiacenza, l'algoritmo di Dijkstra realizzato con un *heap binario* costa: a) $\Theta(n^2 \log n)$ b) $\Theta(m + n\log n)$ c) $\Theta(n^2)$ d) $O(n \log n)$
14. Dato un grafo connesso con n vertici ed m archi rappresentato tramite liste di adiacenza, l'algoritmo di Dijkstra realizzato con una *lista lineare non ordinata* costa: a) $O(mn)$ b) $\Theta(n+m)$ c) $\Theta(m \log n)$ d) $O(m \log n)$
15. Sia d^k_{xy} il costo di un cammino minimo k-vincolato da x a y, secondo la definizione di Floyd e Warshall. Risulta:
    a) $d^k_{xy} = \min\{d^{k-1}_{xy}, d^{k-1}_{xv_k} + d^{k-1}_{v_k x}\}$ b) $d^k_{xy} = \min\{d^{k-1}_{xy}, d^{k-1}_{xv_k} + d^{k-1}_{v_k y}\}$
    c) $d^k_{xy} = \min\{d^{k-1}_{xy}, d^{k}_{xv_k} + d^{k}_{v_k y}\}$ d) $d^k_{xy} = \min\{d^{k}_{xy}, d^{k-1}_{xv_k} + d^{k-1}_{v_k y}\}$
16. L'operazione *Union*(A,B) di 2 insiemi disgiunti A, B con alberi *QuickFind* senza l'euristica dell'unione pesata costa nel caso peggiore: a) $\Theta(\min(|A|,|B|))$ b) $\Theta(\max(|A|,|B|))$ c) $\Theta(|A|)$ d) $\Theta(|B|)$
17. L'operazione *Find*(x) con alberi *QuickUnion* con l'euristica dell'unione pesata *by rank* costa:
    a) $\Theta(n)$ b) $\Theta(1)$ c) $\Theta(\log n)$ d) $O(\log n)$
18. Dato un grafo pesato con n vertici ed m archi, l'algoritmo di Kruskal esegue un numero di operazioni Union(u,v) pari a:
    a) $\Theta(m)$ b) $\Theta(n)$ c) $\Theta(m \log n)$ d) $\Theta(\log n)$
19. Dato un grafo connesso con n vertici ed m archi, l'algoritmo di Prim esegue un numero di operazioni di decremento delle chiavi pari a:
    a) $O(m)$ b) $\Theta(m)$ c) $O(n)$ d) $\Theta(n)$
20. Dato un grafo pesato con n vertici ed m archi, il costo di una fase dell'algoritmo di Borůvka è pari a:
    a) $O(m)$ b) $O(n)$ c) $\Theta(m + n\log n)$ d) $\Theta(m \log n)$

Griglia Risposte

	Domanda																			
Risposta	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
a																				
b																				
c																				
d																				

ESERCIZIO 2 (5 punti) (**Da svolgere sul retro della pagina!**)

La *somma* di 2 grafi $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$ è un grafo $G = (V, E)$ in cui $V = V_1 \cup V_2$, ed $E = E_1 \cup E_2 \cup \{(x,y) | x \in V_1, y \in V_2\}$. Sia G il grafo ottenuto sommando un ciclo di 4 nodi ed un grafo connesso di 2 nodi. Numerare in modo arbitrario i vertici di G da 1 a 6, e pesare ogni arco come somma dei numeri associati ai vertici incidenti. Restituire quindi il minimo albero ricoprente di G, mostrando l'esecuzione passo per passo dell'algoritmo di Kruskal.