Università Degli Studi Di L'Aquila

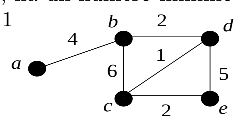
Prova di recupero di **Algoritmi e Strutture Dati**

Mercoledì 21 Marzo 2007 – Prof. Guido Proietti

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

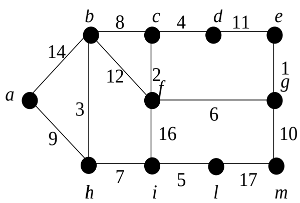
ESERCIZIO 1 (25 punti): Domande a risposta multipla

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. L'algoritmo più efficiente per il calcolo dell' n-esimo numero della sequenza di Fibonacci ha complessità
   a) $\Omega(n)$ b) $\Theta(n)$ *c) $O(\log n)$ d) $\Theta(n \log n)$
2. $f(n) = \Theta(n)$ se e solo se:
   a) $f(n) = O(n)$ e $f(n) = \omega(n)$ *b) $f(n) = O(n)$ e $f(n) = \Omega(n)$ c) $f(n) = o(n)$ e $f(n) = \omega(n)$ d) $f(n) = o(n)$ e $f(n) = \Omega(n)$
3. Quale delle seguenti relazioni asintotiche è vera:
   a) $n \log^2 n = O(n \log n^2)$ b) $n = \Theta(4^{\log n})$ c) $2^{n+1} = \omega(2^n)$ *d) $n \log n^2 = \Theta(n \log n)$
4. Il numero di foglie dell'albero di decisione di un qualsiasi algoritmo per il problema della ricerca in un insieme ordinato è:
   a) $\Theta(n \log n)$ b) $\Theta(\log n)$ *c) $\Omega(n!)$ d) $O(n!)$
5. L'algoritmo di ordinamento non crescente Insertion Sort applicato ad una sequenza di input ordinata in modo non crescente esegue un numero di confronti tra elementi pari a:
   *a) $n-1$ b) n c) $n+1$ d) $n(n-1)/2$
6. L'algoritmo ottimale di fusione di due sequenze ordinate di lunghezza n e n^2 rispettivamente, ha complessità:
   a) $\Theta(n)$ b) $O(n)$ c) $\omega(n^2)$ *d) $\Theta(n^2)$
7. Durante l'esecuzione del Quicksort, applicando la procedura di partizione *in loco* al vettore $[28, 47, 12, 98, 20, 6, 32]$, con perno l'elemento 28, si ottiene
   *a) $[20, 6, 12, 28, 98, 47, 32]$ b) $[12, 6, 20, 28, 98, 47, 32]$ c) $[6, 12, 20, 28, 47, 98, 32]$ d) $[6, 12, 20, 28, 32, 47, 98]$
8. Qual è la complessità spaziale dell'algoritmo Integer Sort applicato ad un array A di n elementi in cui $A[i] = 2i^2 + i$ per $i = 1, .., n$?
   a) $\Theta(n^3)$ b) $\Theta(n)$ *c) $O(n^2)$ d) $\Theta(n \log n)$
9. La procedura *FixHeap*(A, i) per il mantenimento di un heap nel caso migliore costa:
   a) $\Theta(\log n)$ b) $\Omega(\log n)$ c) $\Theta(n)$ *d) $O(1)$
10. Dato un nodo v di un albero AVL di altezza h, sia $\ell(v)$ l'altezza del sottoalbero sinistro di v, e sia $r(v)$ l'altezza del sottoalbero destro di v. Quale delle seguenti espressioni rappresenta il fattore di bilanciamento di v:
    a) $h - \ell(v)$ b) $h - r(v)$ *c) $|\ell(v) - r(v)|$ d) $r(v) - \ell(v)$
11. Dati due elementi u, v appartenenti ad un universo totalmente ordinato U, una funzione hash $h(\cdot)$ si dice *perfetta* se:
    a) $u = v \Rightarrow h(u) \neq h(v)$ b) $u \neq v \Rightarrow h(u) = h(v)$ c) $u = v \Rightarrow h(u) = h(v)$ *d) $u \neq v \Rightarrow h(u) \neq h(v)$
12. Qual è la distanza tra le stringhe *tredici* e *sedici*?
    a) 0 b) 1 *c) 2 d) 3
13. Una coda di priorità realizzata con una lista lineare ordinata supporta l'estrazione del massimo in:
    *a) $O(1)$ b) $\Theta(n)$ c) $\Theta(\log n)$ d) $\Omega(\log n)$
14. Un grafo *non connesso* di n vertici, ha un numero minimo di archi pari a:
    *a) 0 b) $n-1$ c) $n-2$ d) 1
15. La visita in ampiezza del grafo  eseguita partendo dal nodo a genera un albero BFS di altezza pari a:
    a) 1 b) 2 *c) 3 d) 4
16. L'algoritmo di Bellman e Ford applicato ad un grafo pesato con un numero di archi $m = \Theta(n \log n)$, ha complessità:
    a) $\Theta(n^2)$ b) $\Theta(n+m)$ c) $\Theta(n^3)$ *d) $O(n^2 \log n)$
17. Dato un grafo pesato e completo con n vertici, 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) $O(n \log n)$
18. 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) $O(n^2 \log n)$
19. Il *minimo albero ricoprente* del grafo della domanda (15) ha peso totale:
    *a) 9 b) 13 c) 14 d) 5
20. Dato un grafo pesato con n vertici ed m archi, l'algoritmo di Boruvka ha una complessità pari a:
    a) $\Theta(m)$ b) $\Theta(n)$ 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!**)

Mostrare l'intera esecuzione, passo per passo, dell'algoritmo di Dijkstra sul seguente grafo, assumendo il nodo f quale nodo sorgente.