Università degli Studi dell'Aquila

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

Martedì 9 Settembre 2008 – 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 × 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 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. La delimitazione inferiore al problema dell'ordinamento ottenibile dagli alberi di decisione è:
   a) $o(n \log n)$ b) $\omega(n \log n)$ *c) $\Theta(n \log n)$ d) $\Theta(n)$
7. Siano $f(n)$ e $g(n)$ i costi degli algoritmi Heapsort e Quicksort, rispettivamente. Quale delle seguenti relazioni asintotiche è vera:
   a) $g(n) = o(f(n))$ b) $f(n) = \Theta(g(n))$ c) $f(n) = \omega(g(n))$ *d) $g(n) = \omega(f(n))$
8. Sia dato un array A di n elementi in cui l'elemento massimo è pari a n^3. Qual è la complessità temporale dell'algoritmo Radix Sort applicato ad A?
   a) $\Theta(n^3)$ b) $\Theta(n \log_3 n)$ *c) $O(n)$ d) $\Theta(n \log n)$
9. Una coda di priorità realizzata con un heap binario avente nella radice l'elemento minimo, supporta la ricerca del secondo elemento più piccolo in:
   a) $\Theta(n \log n)$ b) $\Theta(n)$ c) $\Theta(\log n)$ *d) $O(1)$
10. Un heap binomiale di 13 elementi è costituito dai seguenti alberi binomiali:
    a) B_0, B_1, B_3 b) B_0, B_1, B_2 *c) B_0, B_2, B_3 d) B_1, B_2, B_3
11. La procedura Heapify per il mantenimento di un heap, nel caso migliore costa:
    a) $\Theta(\log n)$ b) $\Theta(n)$ *c) $\Theta(1)$ d) $\Theta(n \log n)$
12. In un albero AVL di n elementi, la ricerca di un elemento ha complessità:
    *a) $O(\log n)$ b) $\Omega(n)$ c) $\Theta(\log n)$ d) $\Theta(1)$
13. 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) $u \neq v \Rightarrow h(u) \neq h(v)$
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. 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) $O(m \log n)$
16. Dato un grafo pesato con n vertici ed m archi, l'algoritmo di Dijkstra realizzato con heap di Fibonacci ha complessità:
    a) $\Theta(n^2 \log n)$ *b) $\Theta(m + n \log n)$ c) $\Theta(n^2)$ d) $O(n \log n)$
17. Dato un grafo pesato con n vertici ed m archi, l'algoritmo di Floyd e Warshall ha complessità:
    a) $\Theta(n^2)$ b) $\Theta(n+m)$ *c) $O(n^3)$ d) $O(n\,m)$
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. Dato un grafo completo con n vertici, l'algoritmo di Boruvka ha una complessità pari a:
    a) $\Theta(m)$ b) $\Theta(n)$ c) $\Theta(m + n \log n)$ *d) $\Theta(n^2 \log n)$
20. Dato un grafo pesato con n vertici ed m archi, l'algoritmo di Kruskal esegue un numero di operazioni Find(u) pari a:
    a) m b) $\Theta(n)$ c) $\Theta(m \log n)$ *d) $\Theta(m)$

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!**)

Sia $G = (V, E)$ un grafo costituito da 7 vertici, numerati da 1 a 7, e tale che $e = (i, j) \in E$ se e solo se $i + j$ è un numero primo. Si assegni quindi all'arco (i, j) un peso pari a $i + j$, e si mostri l'esecuzione passo per passo dell'algoritmo di Kruskal.