Università degli Studi dell'Aquila

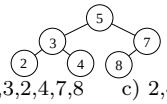
Prova di recupero di **Algoritmi e Strutture Dati**

Martedì 6 Settembre 2011 – 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) $O(\log n)$ b) $O(1)$ c) $\omega(\log n)$ d) $\Theta(n \log n)$
2. Un algoritmo con complessità temporale $O(1)$ su un input di dimensione n esegue un numero di operazioni:
   *a) costante, indipendente da n b) pari ad 1 c) pari ad n, a meno di costanti additive d) pari a 0
3. Dato un problema con una delimitazione inferiore alla complessità temporale pari a $\Omega(f(n))$, un algoritmo per la sua risoluzione non può avere tempo di esecuzione $g(n)$ pari a:
   a) $g(n) = \Theta(f(n))$ *b) $g(n) = o(f(n))$ c) $g(n) = \omega(f(n))$ d) $g(n) = O(f(n))$
4. L'altezza dell'albero di decisione associato al problema dell'ordinamento è:
   *a) $\Omega(n \log n)$ b) $\omega(n \log n)$ c) $O(n \log n)$ d) $\Theta(n!)$
5. Siano $f(n)$ e $g(n)$ i costi dell'algoritmo Insertion Sort nel caso peggiore e in quello migliore, rispettivamente. Quale delle seguenti relazioni asintotiche è vera:
   a) $f(n) = \Omega(g(n))$ *b) $f(n) = \omega(g(n))$ c) $g(n) = \omega(f(n))$ d) $f(n) = o(g(n))$
6. L'algoritmo Merge Sort, applicato ad una sequenza di 2^k elementi, esegue un numero di chiamate ricorsive pari a:
   a) 2^k b) 2^{k-1} *c) $2^{k+1} - 2$ d) 2
7. Siano $f(n)$ e $g(n)$ i costi degli algoritmi Heapsort e Quicksort nel caso peggiore, 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^c, con c costante positiva. Qual è la complessità temporale dell'algoritmo Radix Sort applicato ad A?
   a) $\Theta(n^c)$ b) $\Theta(n \log_c n)$ *c) $O(n)$ d) $\Theta(n \log n)$
9. Per $n = 2^k$, la soluzione dell'equazione di ricorrenza $T(n) = 2 \cdot T(n/4) + n, T(1) = \Theta(1)$, è:
   a) $\Theta(n^2)$ b) $\Theta(n^{\log n})$ *c) $\Theta(n)$ d) $\Theta(n \log n)$
10. L'algoritmo *Heapify* per la costruzione di un heap(A) nel caso peggiore costa:
    a) $\Theta(n \log n)$ b) $\Omega(n \log n)$ c) $O(1)$ *d) $O(n)$
11. Una coda di priorità realizzata con un array ordinato supporta l'estrazione del massimo in:
    a) $\Theta(\log n)$ b) $\Omega(\log n)$ c) $\Theta(n)$ *d) $\Theta(1)$
12. In un heap binomiale di n elementi, la cancellazione di un elemento ha complessità:
    *a) $O(\log n)$ b) $O(1)$ c) $\Theta(n)$ d) $\Theta(1)$
13. Dato l'albero binario 

la visita in ordine simmetrico restituisce:
    *a) 2,3,4,5,8,7 b) 5,3,2,4,7,8 c) 2,4,3,8,7,5 d) 8,7,5,4,3,2
14. In un albero AVL di n elementi, la ricerca di un elemento nel caso migliore costa:
    *a) $O(1)$ b) $\Theta(n)$ c) $\Theta(\log n)$ d) $\Theta(n/\log n)$
15. Quanti archi vanno aggiunti ad un albero di n nodi per renderlo un grafo completo?
    a) 1 b) nessuno *c) $\frac{(n-2)(n-1)}{2}$ d) n
16. L'albero DFS (ovvero ottenuto mediante una *visita in profondità*) di un grafo completo di n vertici ha altezza:
    *a) $n-1$ b) $O(1)$ c) $O(\log n)$ d) 1
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. Sia d_{xy}^k il costo di un cammino minimo k-vincolato da x a y, secondo la definizione di Floyd e Warshall. Risulta:
    a) $d_{xy}^k = \min\{d_{xy}^{k-1}, d_{xv_k}^{k-1} + d_{v_k x}^{k-1}\}$ *b) $d_{xy}^k = \min\{d_{xy}^{k-1}, d_{xv_k}^{k-1} + d_{v_k y}^{k-1}\}$
    c) $d_{xy}^k = \min\{d_{xy}^{k-1}, d_{xv_k}^{k} + d_{v_k y}^{k}\}$ d) $d_{xy}^k = \min\{d_{xy}^{k}, d_{xv_k}^{k-1} + d_{v_k y}^{k-1}\}$
19. In un grafo completo di 7 nodi etichettati da 1 a 7, e tale che l'arco (i, j) ha peso $\max\{i, j\}$, un *minimo albero ricoprente* ha peso:
    a) 6 *b) 27 c) 0 d) 7
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!**)

Mostrare l'intera esecuzione, passo per passo, dell'algoritmo di Kruskal su un grafo completo di 5 nodi etichettati da 1 a 5, e tale che l'arco (i, j) abbia peso $|i - j|$.