Università degli Studi dell'Aquila

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

Martedì 21 Settembre 2010 – 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. Detto F_n l' n-esimo numero della sequenza di Fibonacci, quale delle seguenti relazioni asintotiche è falsa?
   a) $F_n = O(2^n)$ *b) $F_n = \Theta(2^n)$ c) $F_n = \Theta(\phi^n)$ d) $F_n = \Omega(\phi^n)$
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 funzioni $f(n)$ è $\Theta(n)$:
   a) $f(n) = n/\log n$ *b) $f(n) = n + \log n$ c) $f(n) = 1$ d) $f(n) = n^2$
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. Per $n = 2^k$, quale delle seguenti equazioni di ricorrenza descrive più precisamente la complessità dell'algoritmo Merge Sort:
   a) $T(n) = 2 \cdot T(n/2) + n, T(1) = \Theta(1)$ *b) $T(n) = 2 \cdot T(n/2) + \Theta(n), T(1) = 1$ c) $T(n) = 2 \cdot T(n/2) + O(n), T(1) = \Theta(1)$
   d) $T(n) = 2 \cdot T(n/2) + n, T(1) = 1$
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. Qual è la complessità spaziale dell'algoritmo Integer Sort applicato ad un array A di n elementi in cui $A[i] = 3i^3 - 2i^2$ per $i = 1, .., n$?
   *a) $\Theta(n^3)$ b) $\Theta(n)$ c) $O(n^2)$ d) $\Theta(n \log n)$
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. La procedura *FixHeap* per il mantenimento di un heap, nel caso migliore costa:
   a) $\Theta(\log n)$ b) $\Theta(n)$ *c) $\Theta(1)$ d) $\Theta(n \log n)$
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. Sia H_1 un heap binomiale costituito dagli alberi binomiali $\{B_0, B_1, B_5\}$, e sia H_2 un heap binomiale costituito dagli alberi binomiali $\{B_2, B_3, B_5\}$. Da quali alberi binomiali è formato l'heap binomiale ottenuto dalla fusione di H_1 e H_2?
    *a) $\{B_0, B_1, B_2, B_3, B_6\}$ b) $\{B_0, B_1, B_2, B_3, B_4, B_5\}$ c) $\{B_0, B_1, B_2, B_3, B_5\}$ d) $\{B_0, B_1, B_2, B_3, B_5, B_5\}$
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. In un albero binario di ricerca con n elementi e di altezza h, il *predecessore* di un elemento può essere determinato in:
    *a) $O(h)$ b) $O(\log n)$ c) $\Theta(1)$ d) $\Theta(n)$
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. 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
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. 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) $O(m \log n)$
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. 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) $\Theta(|B|)$
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!**)

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 grafo completo di 3 nodi ed un ciclo di lunghezza 4. Numerare in modo arbitrario i vertici di G da 1 a 7, 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 Prim con sorgente il nodo 1.