Università degli Studi dell'Aquila

Prova di Recupero di **Algoritmi e Strutture Dati con Laboratorio**

Martedì 20 Luglio 2010 – Proff. Guido Proietti e Giovanna Melideo

Scrivi i tuoi dati $\Longrightarrow$	Cognome:	Nome:	Matricola: .....

ESERCIZIO 1 (Teoria): Domande a risposta multipla

Premessa: Questa parte è costituita da 10 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 30. Se tale somma è negativa, verrà assegnato 0.

1. $f(n) = \Theta(n)$ 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 $f(n) = \Omega(n)$
2. 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) $O(n!)$
3. 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) $T(n) = 2 \cdot T(n/2) + n, T(1) = 1$
4. 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) $\Theta(n \log n)$
5. 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) $\{B_0, B_1, B_2, B_3, B_5, B_5\}$
6. 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) $\Theta(n)$
7. 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
8. 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) $O(m \log n)$
9. 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) $\Theta(|B|)$
10. 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
a										
b										
c										
d										