Università degli Studi dell'Aquila

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

Martedì 25 Settembre 2012 – Proff. Guido Proietti e Giovanna Melideo

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

ESERCIZIO 1: 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. Se tale somma è negativa, verrà assegnato 0.

1. A quale algoritmo di ordinamento corrisponde il seguente pseudocodice:
   Sort(A)
   for ($k = 0$) to $n - 2$ do
   $m = k + 1$
   for ($j = k + 2$) to n do
   if ($A[j] < A[m]$) then $m = j$
   scambia $A[m]$ con $A[k + 1]$

   *a) Selection Sort b) Insertion Sort c) Quicksort d) L'algoritmo non ordina.

2. Quale delle seguenti implicazioni è falsa:
   a) $f(n) = \Theta(g(n)) \Rightarrow f(n) = O(g(n))$ *b) $f(n) = O(g(n)) \Rightarrow f(n) = o(g(n))$ c) $f(n) = \Theta(g(n)) \Rightarrow g(n) = \Omega(f(n))$ d) $f(n) = o(g(n)) \Rightarrow g(n) = \omega(f(n))$

3. L'algoritmo ottimale di fusione di due sequenze ordinate di lunghezza p e q rispettivamente, ha complessità:
   a) $\Theta(p \cdot q)$ b) $\Theta(p)$ c) $\omega(p + q)$ *d) $\Theta(p + q)$

4. 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) $T(n) = 2 \cdot T(n/2) + n, T(1) = 1$

5. La procedura $FixHeap(A, 2)$ per il mantenimento di un heap applicata al vettore $A = [12, 5, 2, 6, 9, 3]$ restituisce:
   a) $A = [12, 9, 3, 6, 5, 2]$ b) $A = [12, 6, 2, 5, 9, 3]$ c) $A = [12, 5, 3, 6, 9, 2]$ *d) $A = [12, 9, 2, 6, 5, 3]$

6. Una coda di priorità realizzata con un array non ordinato supporta l'estrazione del massimo in:
   a) $\Theta(\log n)$ b) $O(\log n)$ c) $\Theta(1)$ *d) $\Theta(n)$

7. In un albero AVL di n elementi, la cancellazione della radice comporta nel caso migliore un numero di rotazioni pari a:
   *a) 0 b) 2 c) $\Theta(\log n)$ d) 1

8. 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) $u \neq v \Rightarrow h(u) \neq h(v)$

9. L'algoritmo di Bellman e Ford applicato ad un grafo pesato con un numero di archi $m = \Theta(n^2)$, ha complessità:
   a) $\Theta(n^2)$ b) $\Theta(n + m)$ *c) $\Theta(n^3)$ d) $O(m \log n)$

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) $O(m)$ b) $\Theta(n)$ c) $\Theta(m \log n)$ d) m

Griglia Risposte

	Domanda									
Risposta	1	2	3	4	5	6	7	8	9	10
a										
b										
c										
d										