Università degli Studi dell'Aquila

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

Venerdì 21 Giugno 2013 – Proff. Guido Proietti e Giovanna Melideo

Scrivi i tuoi dati ⟹	Cognome: ..... Nome: ..... Matricola: .....

PROVA SCRITTA DI TEORIA: Domande a risposta multipla

Premessa: La prova 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 finale della prova è ottenuto sommando i punti attribuiti e normalizzando su base 30. Se tale somma è negativa, verrà assegnato 0.

1. 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) esattamente $n-1$ b) almeno n c) esattamente $n+1$ d) almeno $n(n-1)/2$
2. Siano $f(n)$ e $g(n)$ i costi dell'algoritmo Selection Sort nel caso peggiore e in quello migliore, rispettivamente. Quale delle seguenti relazioni asintotiche è vera:
   a) $f(n) = \omega(g(n))$ *b) $f(n) = \Theta(g(n))$ c) $g(n) = \omega(f(n))$ d) $f(n) = o(g(n))$
3. Quale tra i seguenti rappresenta lo pseudocodice dell'algoritmo Heapsort:

   *a)
   ```
   Heapsort(A)
   Heapify(A)
   heapsize[A] = n
   for (i = n) down to 2 do
     scambia A[1] con A[i]
     heapsize[A]=heapsize[A] − 1
     Fixheap(1, A)
   ```

   b)
   ```
   Heapsort(A)
   Heapify(A)
   heapsize[A] = n
   for (i = n) down to 2 do
     scambia A[n] con A[i]
     heapsize[A]=heapsize[A] − 1
     Fixheap(1, A)
   ```

   c)
   ```
   Heapsort(A)
   Heapify(A)
   heapsize[A] = n
   for (i = n) down to 1 do
     scambia A[1] con A[i]
     heapsize[A]=heapsize[A] − 1
     Fixheap(1, A)
   ```

   d)
   ```
   Heapsort(A)
   Heapify(A)
   heapsize[A] = n
   for (i = n) down to 2 do
     scambia A[1] con A[i]
     heapsize[A]=heapsize[A] − 1
     Fixheap(n, A)
   ```
4. Dato un heap binomiale di n elementi, l'inserimento di $k = \Theta(n)$ elementi in successione costa:
   a) $O(\log n)$ b) $O(k)$ c) $\Theta(n)$ *d) $O(n \log n)$
5. Dato un albero AVL di n elementi, la cancellazione in successione di un numero costante k di elementi può essere eseguita in:
   a) $\Theta(n)$ *b) $O(\log n)$ c) $\Theta(k)$ d) $O(n)$
6. Si consideri il grafo $G = (V, E)$ con $V = \{1, 2, 3, 4, 5, 6\}$ ed $E = \{(1, 4), (2, 4), (2, 5), (3, 5)\}$. Quali delle seguenti affermazioni è <u>falsa</u>:
   a) G è disconnesso b) la distanza in G tra il nodo 3 e il nodo 4 è 3 *c) il *diametro* di G, ovvero la distanza massima tra due nodi in G, è pari a 4 d) G ha grado 2
7. La visita in ampiezza del grafo di cui alla domanda precedente eseguita partendo dal nodo 2 restituisce un BFS di altezza:
   a) 1 *b) 2 c) 3 d) 4
8. Dato un grafo pesato sugli archi con pesi reali positivi e avente n vertici ed $m = \Theta(n \log n)$ archi, l'algoritmo di Dijkstra realizzato con heap binomiali costa:
   a) $\Theta(n^2)$ b) $\Theta(n + m)$ c) non si può applicare *d) $O(n \log^2 n)$
9. Dato un grafo completo con n vertici, l'algoritmo di Prim eseguito con un *array lineare non ordinato* costa:
   a) $o(n^2)$ b) $\Theta(n^3)$ c) $\Theta(n^2 \log n)$ *d) $\Theta(n^2)$
10. Dato un grafo connesso di n nodi ed m archi, per quali valori (asintotici) di m si ha che l'implementazione di Prim con heap di Fibonacci <u>non è</u> strettamente più efficiente dell'implementazione di Kruskal con alberi QuickUnion con euristica di bilanciamento *union by size*?
    a) $m = \Theta(n \log n)$ b) tutti c) nessuno *d) $m = \Theta(n)$

Griglia Risposte

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