Università degli Studi dell'Aquila

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

Martedì 10 Settembre 2019 – Prof. Guido Proietti (Modulo di Teoria)

Scrivi i tuoi dati ⟹	Cognome:	Nome:	Matricola:	PUNTI
ESERCIZIO 1	Risposte Esatte:	Risposte Omesse:	Risposte Errate:	

ESERCIZIO 1 (Domande a risposta multipla): 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 × 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 30. Se tale somma è negativa, verrà assegnato 0.

1. Siano $f(n)$ e $g(n)$ i costi dell'algoritmo InsertionSort2 nel caso migliore e Quicksort in quello medio, rispettivamente. Quale delle seguenti relazioni asintotiche è vera:
   *a) $f(n) = o(g(n))$ b) $f(n) = \Theta(g(n))$ c) $f(n) = \omega(g(n))$ d) $f(n) = \Omega(g(n))$
2. Siano $f(n)$ e $g(n)$ i costi dell'algoritmo Selection Sort nel caso migliore e in quello medio, rispettivamente. Quale delle seguenti relazioni asintotiche è vera:
   *a) $f(n) = \Omega(g(n))$ b) $f(n) = \omega(g(n))$ c) $g(n) = o(f(n))$ d) $f(n) = o(g(n))$
3. Sotto quali ipotesi la soluzione dell'equazione di ricorrenza $T(n) = a \cdot T(n/b) + f(n)$, con $T(1) = \Theta(1)$, a, b costanti non negative, è pari a $T(n) = \Theta(n^{\log_b a})$?
   a) Se $f(n) = O\left(n^{\log_b a+\epsilon}\right)$, per qualche $\epsilon > 0$, e se vale la condizione di regolarità: $af(n/b) \leq cf(n)$ per qualche $c < 1$ ed n sufficientemente grande
   *b) Se $f(n) = O\left(n^{\log_b a-\epsilon}\right)$, per qualche $\epsilon > 0$
   c) Se $f(n) = \Theta\left(n^{\log_b a}\right)$ d) Se $f(n) = \Omega\left(n^{\log_b a-\epsilon}\right)$, per qualche $\epsilon > 0$
4. Quale tra i seguenti rappresenta lo pseudocodice dell'algoritmo Heapsort:

   a)
   ```
   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)
   ```

   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 2 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)
   ```
5. Quali sono, rispettivamente, i costi per implementare le operazioni di *IncreaseKey*, *DecreaseKey*, e *Merge* in una coda di priorità di n elementi implementata utilizzando un d-heap?
   a) $O(d \log_d n), O(d \log_d n), \Theta(n)$ b) $O(\log_d n), O(\log_d n), O(n)$ *c) $O(d \log_d n), O(\log_d n), \Theta(n)$ d) $O(n), O(\log_d n), \Theta(d \log_d n)$
6. Dato un albero AVL T contenente n elementi, si consideri l'inserimento di una sequenza di 2^n elementi in T. L'altezza dell'AVL risultante sarà:
   *a) $\Theta(n)$ b) $\Theta(n^2)$ c) $\Theta(\log^2 n)$ d) $\Theta(\log n)$
7. In un grafo *completo* e non pesato con n vertici, il cammino semplice di lunghezza **massima** tra due vertici fissati è lungo:
   a) 1 b) 2 c) $n(n-1)/2$ *d) $n-1$
8. L'algoritmo di Bellman e Ford applicato ad un grafo pesato con un numero di archi $m = \Theta(n \log n)$, ha complessità:
   a) $\Theta(n^2)$ b) $\Theta(n+m)$ c) $\Theta(n^3)$ *d) $O(n^2 \log n)$
9. Dato un grafo connesso di n nodi ed m archi, per quale valore (asintotico) di m si ha che l'implementazione dell'algoritmo di Dijkstra con heap di Fibonacci ha la stessa complessità temporale dell'implementazione con heap binomiali?
   *a) $m = \Theta(n)$ b) $m = \Theta(n^2)$ c) $m = \Theta(n \log n)$ d) mai
10. Si consideri l'algoritmo di Prim per il calcolo del minimo albero ricoprente di un grafo connesso con n nodi ed m archi. Quale delle seguenti opzioni descrive in modo preciso il numero di operazioni di *Insert*, *deleteMin*, e *decreaseKey* eseguite dall'algoritmo?
    a) $O(n), O(1), \Theta(n)$ b) $\Theta(n), n, O(n)$ c) $O(n), O(n), \Theta(m)$ *d) $n, n, O(m)$

Griglia Risposte

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