Università degli Studi dell'Aquila

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

Martedì 18 Febbraio 2014 – 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

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 × 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. Quale delle seguenti relazioni di ricorrenza descrive la complessità dell'algoritmo più efficiente per il calcolo della sequenza di Fibonacci basato sul prodotto di matrici?
   a) $T(n) = 2T(n/2) + O(1)$ se $n \geq 2$, $T(1) = O(1)$ se $n = 1$ b) $T(n) = 2T(n/4) + O(1)$ se $n \geq 2$, $T(1) = O(1)$ se $n = 1$
   *c) $T(n) = T(n/2) + O(1)$ se $n \geq 2$, $T(1) = O(1)$ se $n = 1$ d) $T(n) = 2T(n/2) + O(1)$ se $n \geq 2$, $T(1) = O(n)$ se $n = 1$
2. Sia dato un array A di n elementi in cui l'elemento massimo è pari a $\Theta(n^4)$. Quante passate di Bucket Sort sono necessarie all'algoritmo Radix Sort per ordinare A?
   a) 1 b) $\Theta(n)$ *c) $\Theta(1)$ d) $\Theta(\log n)$
3. Quale tra i seguenti rappresenta lo pseudocodice dell'algoritmo Heapsort:

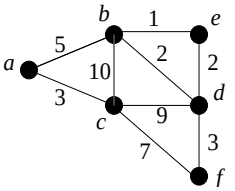
a)
```
Heapsort(A)
Heapify(A)
heapsize[A] = n
for (i = n) down to 1 do
  scambia A[1] con A[i]
  heapsize[A]=heapsize[A] − 1
  Fixheap(1, A)
```

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)
```

4. Dato un albero AVL T di altezza $\Theta(\log n)$, si consideri la cancellazione di una sequenza di $\Theta(n)$ elementi da T. L'altezza dell'AVL risultante sarà:
   a) 0 b) $O(1)$ *c) $O(\log n)$ d) $\Theta(\log n)$



5. Si consideri il grafo G in figura. Quale delle seguenti affermazioni è vera?
   a) G è bipartito *b) G contiene sottografi indotti completi
   c) G è regolare d) Tutti i sottografi indotti di G sono aciclici
6. Si consideri il grafo di cui alla domanda (5) e si orientino gli archi dal nodo con lettera maggiore al nodo con lettera minore secondo l'ordine alfabetico. Quanti rilassamenti esegue in totale alla fine della prima passata l'algoritmo di Bellman e Ford con sorgente d e con l'ipotesi che gli archi vengano considerati in ordine lessicografico?
   a) 0 b) 1 *c) 2 d) 4
7. Si consideri il grafo di cui alla domanda (5) e si numerino i vertici nel seguente modo: $a := 1; b := 2; c := 3, d := 4; e := 5; f := 6$. Si orientino ora gli archi dal nodo con numero minore al nodo con numero maggiore, e si supponga di applicare l'algoritmo di Floyd e Warshall. Quanti rilassamenti vengono eseguiti durante la seconda passata dell'algoritmo?
   a) 0 b) 1 c) 2 *d) 3
8. Si consideri il grafo di cui alla domanda (5) e si orientino gli archi dal nodo con lettera maggiore al nodo con lettera minore secondo l'ordine alfabetico. Quali tra i seguenti è un ordinamento topologico del grafo ottenuto?
   a) $\langle b, a, e, c, f, d\rangle$ b) $\langle a, b, e, d, c, f\rangle$ *c) $\langle a, b, c, d, f, e\rangle$ d) il grafo non è aciclico
9. Si consideri la gestione di n insiemi disgiunti sottoposti ad $n - 1$ operazioni di *Union* e a $\Theta(n^2)$ operazioni di *Find* mediante l'utilizzo di alberi *QuickUnion*. Quanto costa complessivamente la gestione della sequenza di operazioni?
   a) $O(n^2)$ b) $O(n^2 \log n)$ *c) $O(n^3)$ d) $\Theta(n^3)$
10. Dato un grafo connesso di n nodi ed m archi, per quale valore (asintotico) di m si ha che l'implementazione di Prim con array non ordinati ha la stessa complessità temporale dell'algoritmo di Borůvka?
    *a) $m = \Theta(n^2/\log n)$ b) $m = \Theta(n^2)$ c) per ogni valore di m d) per nessun valore di m

Griglia Risposte

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