Università degli Studi dell'Aquila

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

Martedì 20 Giugno 2017 – 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.

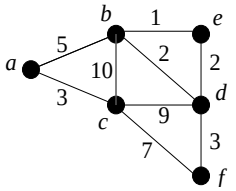
1. Detto F_n l' n-esimo numero della sequenza di Fibonacci, quale delle seguenti proprietà di matrici è vera?
   a) $\begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}^n = \begin{pmatrix}F_n & F_{n+1}\\ F_{n+1} & F_{n-1}\end{pmatrix}$ b) $\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^n = \begin{pmatrix}F_{n+1} & F_n\\ F_n & F_{n-1}\end{pmatrix}$ c) $\begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}^n = \begin{pmatrix}F_{n-1} & F_n\\ F_n & F_{n+1}\end{pmatrix}$ *d) $\begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}^n = \begin{pmatrix}F_{n+1} & F_n\\ F_n & F_{n-1}\end{pmatrix}$
2. Sia dato in input l'array $A = [1, 2, 3, \ldots, n-1, n]$, e si supponga di applicare su di esso gli algoritmi di ordinamento non crescente SelectionSort, InsertionSort1 e InsertionSort2. Quale dei tre algoritmi esegue il minor numero di confronti asintoticamente?
   a) SelectionSort b) InsertionSort1 c) InsertionSort2 *d) Sono tutt'e tre equivalenti
3. Un algoritmo ha una complessità temporale $O(f(n))$ se:
   a) Il tempo di esecuzione $T(n)$ dell'algoritmo su uno specifico input di dimensione n verifica $T(n) = O(f(n))$
   *b) Il tempo di esecuzione $T(n)$ dell'algoritmo su ogni input di dimensione n verifica $T(n) = O(f(n))$
   c) Il tempo di esecuzione medio $T(n)$ dell'algoritmo su un input di dimensione n verifica $T(n) = O(f(n))$
   d) Nel caso migliore, il tempo di esecuzione $T(n)$ dell'algoritmo su un input di dimensione n verifica $T(n) = O(f(n))$
4. Quale tra i seguenti è lo pseudocodice del Merge Sort (si supponga che la chiamata principale sia MergeSort$(A, 1, n)$):

   *a)
   MergeSort(A, i, f)
   if $(i \geq f)$ then return
   $m = (i + f)/2$
   MergeSort(A, i, m)
   MergeSort$(A, m + 1, f)$
   Merge(A, i, m, f)

   b)
   MergeSort(A, i, f)
   if $(i \leq f)$ then return
   $m = (i + f)/2$
   MergeSort(A, i, m)
   MergeSort$(A, m + 1, f)$
   Merge(A, i, m, f)

   c)
   MergeSort(A, i, f)
   if $(i \geq f)$ then return
   $m = (i + f)/2$
   Merge(A, i, m, f)
   MergeSort(A, i, m)
   MergeSort$(A, m + 1, f)$

   d)
   MergeSort(A, i, f)
   if $(i \geq f)$ then return
   $m = (i + f)/2$
   MergeSort(A, i, m)
   MergeSort$(A, m + 1, f)$
   Merge(A, i, f, m)
5. Sia H un heap binomiale costituito dagli alberi binomiali $\{B_0, B_1, B_5\}$, e si supponga di eseguire su di esso un'operazione di *deleteMin()*. Da quali alberi binomiali è formato l'heap binomiale risultante?
   *a) $\{B_1, B_5\}$ b) $\{B_0, B_1, B_5\}$ c) $\{B_0, B_1, B_2, B_3, B_4\}$ d) $\{B_0, B_5\}$
6. Supponiamo di inserire $\sqrt{n}$ elementi in un albero AVL contenente $\log n$ elementi. A quanto sarà pari l'altezza dell'AVL risultante?
   a) $\Theta(\sqrt{n})$ b) $\Theta(\log \log n)$ c) $\Theta(\log(n \cdot \sqrt{n}))$ *d) $\Theta(\log n)$



7. Un grafo si dice *k-vertice-connesso*, con $k \geq 0$, se il numero minimo di vertici che devono essere rimossi (assieme agli archi incidenti) dal grafo per disconnetterlo è pari a k. Per quale valore di k il grafo in figura è k-vertice-connesso?
   a) 0 b) 1 *c) 2 d) 3
8. Si consideri il grafo di cui alla domanda (7) e si supponga di applicare l'algoritmo di Dijkstra per determinare l'albero dei cammini minimi radicato in c. Qual è la sequenza di vertici aggiunti alla soluzione?
   a) $\langle c, a, f, d, b, e\rangle$ b) $\langle c, a, b, f, d, e\rangle$ *c) $\langle c, a, f, b, d, e\rangle$ d) $\langle c, a, b, e, d, f\rangle$
9. Sia dato un grafo connesso G con n vertici, numerati da 1 ad n, ed $m = \Theta(n \log n)$ archi orientati, disposti in modo arbitrario, ma in modo tale da garantire l'aciclicità. Si applichi ora l'algoritmo di ordinamento topologico rispetto al nodo sorgente etichettato 1. La complessità risultante è pari a:
   a) $\Theta(n^2)$ b) $\Theta(n)$ *c) $\Theta(n \log n)$ d) indefinita (non è detto che l'algoritmo possa essere applicato)
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 heap di Fibonacci è <u>meno efficiente</u> dell'implementazione di Kruskal con alberi QuickUnion con euristica di bilanciamento *union by size*?
    a) $m = \Theta(n)$ b) sempre *c) mai d) $m = \omega(n)$

Griglia Risposte

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