

Università degli Studi dell'Aquila

Prima Prova Parziale di Algoritmi e Strutture Dati con Laboratorio

Martedì 2 Dicembre 2014 - Prof. Guido Proietti (Modulo di Teoria)

	Scrivi i tuoi dati \Longrightarrow	Cognome:	Nome:	Matricola:	PUNTI
ſ	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 \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 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 \ge 2, T(1) = O(1) se n = 1 b) T(n) = 2T(n/4) + O(1) se n \ge 2, T(1) = O(1) se n = 1 *c) T(n) = T(n/2) + O(1) se n \ge 2, T(1) = O(1) se n = 1 d) T(n) = 2T(n/2) + O(1) se n \ge 2, T(1) = O(n) se n = 1
```

2. Quale tra i seguenti rappresenta lo pseudocodice dell'algoritmo INSERTIONSORT2 per l'ordinamento non decrescente:

```
InsertionSort(A)
                               INSERTIONSORT(A)
                                                              INSERTIONSORT(A)
                                                                                             INSERTIONSORT(A)
for k = 1 to n - 1 do
                               for k = 1 to n - 1 do
                                                              for k = 1 to n - 1 do
                                                                                             for k = 1 to n - 1 do
 x = A[k+1]
                                x = A[k]
                                                               x = A[k+1]
                                                                                              x = A[k+1]
 j = k
                                j = k - 1
                                                               j = k
                                                                                              j = k
                                while j > 0 e A[j] > x do
A[j+1] = A[j]
                                                                                               while j > 0 e A[j] > x do
 while j > 0 e A[j] > x do
                                                               while j > 0 e A[j] < x do
                                                                                                A[j+1] = A[j]
j = j-1
                                                                 A[j+1] = A[j]
j = j-1
   A[j+1] = A[j]
   j = j - 1
                                  j = j - 1
 A[j] = x
                                A[j+1] = x
                                                                A[j+1] = x
                                                                                               A[j+1] = x
```

3. Siano f(n) e g(n) i costi dell'algoritmo Insertion Sort nel caso peggiore e Selection Sort 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) g(n) = \omega(f(n))
```

4. Siano f(n) e g(n) i costi dell'algoritmo MERGESORT nel caso peggiore e QUICKSORT in quello migliore, 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) g(n) = \omega(f(n))
```

5. Sia dato un array A di n elementi in cui l'elemento massimo è pari a n^3 . Qual è la complessità temporale dell'algoritmo RADIX SORT applicato ad A?

```
a) \Theta(n^3) b) \Theta(n \log_3 n) *c) O(n) d) \Theta(n \log n)
```

6. Quale tra i seguenti rappresenta lo pseudocodice dell'algoritmo HEAPSORT:

```
*c)
                                                                          \overset{\cdot}{\mathrm{HEAPSORT}}(A)
Heapsort(A)
                                     Heapsort(A)
                                                                                                                Heapsort(A)
Heapify(A)
                                     Heapify(A)
                                                                          \text{Heapify}(A)
                                                                                                                Heapify(A)
                                                                          \text{heapsize}[A] = n
heapsize[A] = n
                                     heapsize[A] = n
                                                                                                                heapsize[A] = n
for (i = n) down to 1 do
                                     for (i = n) down to 2 do
                                                                          for (i = n) down to 2 do
                                                                                                                for (i = n) down to 2 do
                                      scambia A[n] con A[i] heapsize [A] = heapsize [A] - 1
 scambia A[1] con A[i]
                                                                            scambia A[1] con A[i]
                                                                                                                  scambia A[1] con A[i]
 heapsize[A] = heapsize[A] - 1
                                                                            heapsize[A] = heapsize[A] - 1
                                                                                                                 heapsize[A] = heapsize[A] - 1
 Fixheap(1, A)
                                       Fixheap(1, A)
                                                                            Fixheap(1, A)
                                                                                                                 Fixheap(n, A)
```

- 7. La procedura Heap-Insert(A, 18) applicata al vettore A = [12, 9, 3, 6, 5, 2, nil] rappresentante un heap binario restituisce: a) A = [18, 9, 12, 6, 5, 3, 2] b) A = [12, 9, 18, 6, 5, 3, 2] c) A = [12, 9, 18, 6, 5, 2, 3] *d) A = [18, 9, 12, 6, 5, 2, 3]
- 8. Sia H un heap binomiale costituito dagli alberi binomiali $\{B_0, B_1, B_2, B_4\}$, e si supponga di eseguire su di esso un'operazione di inserimento. Da quali alberi binomiali è formato l'heap di nomiale risultante?

```
*a) \{B_3, B_4\} b) \{B_0, B_1, B_5\} c) \{B_0, B_1, B_2, B_3, B_4\} d) \{B_5\}
```

9. Dato un albero AVL T contenente n elementi, si consideri l'inserimento di una sequenza di n^2 elementi in T. L'altezza dell'AVL risultante è:

```
a) \Theta(n) b) \Theta(n^2) c) \Theta(\log^2 n) *d) \Theta(\log n)
```

10. Sia dato un albero AVL di altezza h, e si supponga di cancellare un elemento che provochi lo sbilanciamento dell'albero. Quale tra le seguenti è una terna di valori ammissibili per l'altezza dell'AVL prima della cancellazione, dopo la cancellazione, e infine dopo il ribilanciamento complessivo dell'AVL:

a)
$$h, h-1, h-1$$
 b) $h, h, h-2$ c) $h, h-1, h$ *d) $h, h, h-1$

Griglia Risposte

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