I fill a filova i afziale di Algorithii e Strutture Dati con Laboratorio Martedì 5 Novembre 2019 – Prof. Guido Proietti (Modulo di Teoria)

	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 \times 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 x 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 Fibonacci2? 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) = 2 + T(n-1) + T(n-2) se $n \ge 3$, T(1) = T(2) = 1 se n = 1 d) T(n) = 2 + T(n-1) se $n \ge 2$, T(1) = 1se n=1
- 2. Quale delle seguenti relazioni relative alla notazione asintotica è falsa:

```
b) \omega(f(n)) \subseteq \Omega(f(n)) c) O(f(n)) \cap \Omega(f(n)) = \Theta(f(n))
                                                                                                                      *d) o(f(n)) \cap \Theta(f(n)) = \Theta(f(n))
a) o(f(n)) \subseteq O(f(n))
```

3. Si consideri l'algoritmo di ricerca binaria di un elemento in un insieme ordinato di n elementi. Quale delle seguenti opzioni descrive in modo preciso i costi nel caso migliore, peggiore e medio?

```
a) T_{\text{best}}(n) = O(1), T_{\text{worst}}(n) = \Theta(n), T_{\text{avg}}(n) = \Theta(\log n)
                                                                                                   *b) T_{\text{best}}(n) = O(1), T_{\text{worst}}(n) = \Theta(\log n), T_{\text{avg}}(n) = \Theta(\log n)
c) T_{\text{best}}(n) = O(\log n), T_{\text{worst}}(n) = \Theta(n), T_{\text{avg}}(n) = \Theta(\log n)
                                                                                                        d) T_{\text{best}}(n) = O(1), T_{\text{worst}}(n) = O(n), T_{\text{avg}}(n) = \Theta(\log n)
```

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

```
\stackrel{\smile}{\text{InsertionSort}}(A)
InsertionSort(A)
                                                                I_{NSERTIONSORT}(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]
                                 i = k - 1
                                                                 i = k
                                                                                                 i = k
  i = 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]
                                                                   A[j+1] = A[j]
                                                                                                  A[j+1] = A[j]
    j = j - 1
                                   i = i - 1
                                                                   i = i - 1
  A[j] = x
                                 A[i + 1] = x
                                                                 A[j+1] = x
                                                                                                 A[i + 1] = x
```

- 5. 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))
- 6. A quale delle seguenti classi <u>non</u> appartiene la complessità computazionale dell'algoritmo MERGE SORT: b) $O(n^2)$ c) $\Omega(n)$ d) $\Theta(n \log n)$ *a) $o(n \log n)$
- 7. Sia f(n) il costo dell'algoritmo Heapsort nel caso peggiore, e sia g(n) il costo dell'algoritmo Quicksort nel caso migliore. Quale delle seguenti relazioni asintotiche è vera:
 - *b) $f(n) = \Theta(g(n))$ c) $f(n) = \Theta(g(n) \cdot \log n)$ d) $g(n) = \omega(f(n))$ a) g(n) = o(f(n))
- 8. La procedura FixHeap(A, 2) per il mantenimento di un heap applicata al vettore A = [12, 9, 2, 6, 5, 3] restituisce: a) $\hat{A} = \begin{bmatrix} 12, 9, 3, 6, 5, 2 \end{bmatrix}$ b) $\hat{A} = \begin{bmatrix} 12, 6, 2, 5, 9, 3 \end{bmatrix}$ c) $\hat{A} = \begin{bmatrix} 12, 5, 3, 6, 9, 2 \end{bmatrix}$ *d) $\hat{A} = \begin{bmatrix} 12, 9, 2, 6, 5, 3 \end{bmatrix}$
- 9. 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?
 - b) $O(\log_d n), O(\log_d n), O(n)$ *c) $O(d \log_d n), O(\log_d n), \Theta(n)$ d) $O(n), O(\log_d n), \Theta(d \log_d n)$ a) $O(d \log_d n), O(d \log_d n), \Theta(n)$
- 10. Dato un heap binomiale H di n elementi, quale delle seguenti affermazioni è vera:
 - 'a) Il grado della radice di ogni albero in $H \in O(\log n)$, e l'altezza di ogni albero in $H \in O(\log n)$;
 - b) Il grado della radice di ogni albero in $H \in \Theta(\log n)$, e il numero di elementi di qualche albero in $H \in O(\log n)$;
 - c) Il grado della radice di ogni albero in $H \in O(\log n)$, e l'altezza di ogni albero in $H \in O(\log n)$;
 - d) Il grado della radice di ogni albero in $H \in o(\log n)$, e il numero di elementi di ogni albero in $H \in \Theta(\log n)$.

Griglia Risposte

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