

	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  $\times$  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.

- Quali delle seguenti relazioni di ricorrenza descrivono la complessità dell'algoritmo **Fibonacci2**?  
 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)  $T(n) = 2 + T(n-1) + T(n-2)$  se  $n \geq 3$ ,  $T(1) = T(2) = 1$  se  $n = 1$     d)  $T(n) = 2 + T(n-1)$  se  $n \geq 2$ ,  $T(1) = 1$  se  $n = 1$
- 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)  $f(n) = \Omega(g(n))$
- Si consideri l'algoritmo di ricerca binaria di un elemento in un insieme ordinato di  $n$  elementi. Quale delle seguenti opzioni descrive in modo corretto il numero di confronti nel caso migliore, peggiore e medio?  
 a)  $T_{\text{best}}(n) = 1, T_{\text{worst}}(n) = \lfloor \log n \rfloor + 1, T_{\text{avg}}(n) = \Theta(1)$     \*b)  $T_{\text{best}}(n) = 1, T_{\text{worst}}(n) = \lfloor \log n \rfloor + 1, T_{\text{avg}}(n) = \Theta(\log n)$   
 c)  $T_{\text{best}}(n) = 1, T_{\text{worst}}(n) = \lfloor \log n \rfloor + 1, T_{\text{avg}}(n) = \log n$     d)  $T_{\text{best}}(n) = 1, T_{\text{worst}}(n) = \lfloor \log n \rfloor, T_{\text{avg}}(n) = \Theta(n)$
- Per  $n = 2^k$ , la soluzione dell'equazione di ricorrenza  $T(n) = 3 \cdot T(n/2) + n, T(1) = \Theta(1)$ , è:  
 a)  $\Theta(n^{\log_3 2})$     \*b)  $\Theta(n^{\log 3})$     c)  $\Theta(n \log n)$     d)  $\Theta(n)$
- Siano  $f(n)$  e  $g(n)$  i costi degli algoritmi **MERGE-SORT** e **QUICKSORT** nel caso peggiore, 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)  $f(n) = \Omega(g(n))$
- Qual è la complessità temporale dell'algoritmo **RADIX SORT** applicato ad un array  $A$  di  $n$  elementi in cui l'elemento massimo è pari ad  $k = n^4 + n^3 - 3$ ?  
 a)  $\Theta(n^4)$     \*b)  $\Theta(n)$     c)  $O(n^7)$     d)  $\Theta(n \log n)$
- Quale tra i seguenti rappresenta lo pseudocodice dell'algoritmo **HEAPSORT**:  

a)	b)	*c)	d)
HEAPSORT( $A$ )	HEAPSORT( $A$ )	HEAPSORT( $A$ )	HEAPSORT( $A$ )
Heapify( $A$ )	Heapify( $A$ )	Heapify( $A$ )	Heapify( $A$ )
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 $A[1]$ con $A[i]$	scambia $A[n]$ 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	heapsize[ $A$ ] = heapsize[ $A$ ] - 1
Fixheap(1, $A$ )	Fixheap(1, $A$ )	Fixheap(1, $A$ )	Fixheap( $n$ , $A$ )
- Quale tra i seguenti algoritmi non è ottimo se applicato al problema descritto?  
 a) **HEAPSORT** per ordinare una sequenza di  $n$  interi arbitrari  
 \*b) **MERGESORT** per ordinare una sequenza di  $n$  interi con valori compresi tra 1 e  $n^c$   
 c) Algoritmo di ricerca sequenziale per cercare un elemento in una sequenza di  $n$  interi non ordinati  
 d) **INTEGER SORT** per ordinare una sequenza di  $n$  interi con valori  $O(n)$
- Quale tra le seguenti affermazioni è *falsa*?  
 a) Un albero  $d$ -ario quasi completo di  $n$  nodi ha altezza  $\Theta(\log_d n)$   
 b) Un albero binario completo di  $n$  nodi ha esattamente  $\lceil n/2 \rceil$  foglie  
 c) Il numero di nodi interni di un albero strettamente binario è pari al numero di foglie meno 1  
 \*d) Un albero binario di  $n$  nodi ha al più  $\lfloor n/2 \rfloor$  nodi interni
- Sia  $H$  un heap binomiale di 31 elementi, e si supponga che l'elemento minimo nell'heap sia contenuto nell'albero binomiale  $B_3$  di  $H$ . A seguito della cancellazione del minimo da  $H$ , quante operazioni di fusione vengono eseguite nella procedura **ristruttura()**?  
 a) 1    b) 2    \*c) 3    d) 4

## Griglia Risposte

[illegible]