



**UNIVERSITÀ DEGLI STUDI DELL'AQUILA**  
**Prova di recupero di Algoritmi e Strutture Dati**  
 Martedì 12 Luglio 2011 – Prof. Guido Proietti

Scrivi i tuoi dati →	Cognome: .....	Nome: .....	Matricola: .....	<b>PUNTI</b>
<b>ESERCIZIO 1</b>	Risposte Esatte:	Risposte Omesse:	Risposte Errate:	

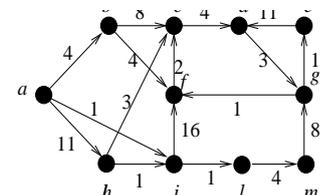
**ESERCIZIO 1 (25 punti): Domande a risposta multipla**

**Premessa:** Questa parte è costituita da 20 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 25. 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. L'algoritmo INSERTION SORT, nel caso migliore esegue un numero di confronti tra elementi pari a:  
 \*a)  $n - 1$     b)  $n$     c)  $n + 1$     d)  $n(n - 1)/2$
3. Quale delle seguenti funzioni  $f(n)$  è  $\Theta(n)$ :  
 a)  $f(n) = n/\log n$     \*b)  $f(n) = n + \log n$     c)  $f(n) = 1$     d)  $f(n) = n^2$
4. 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  $T(n) = O(f(n))$
5. Siano  $f(n)$  e  $g(n)$  i costi dell'algoritmo SELECTION SORT nel caso migliore e in quello peggiore, rispettivamente. Quale delle seguenti relazioni asintotiche è vera:  
 \*a)  $f(n) = \Omega(g(n))$     b)  $f(n) = \omega(g(n))$     c)  $g(n) = \omega(f(n))$     d)  $f(n) = o(g(n))$
6. L'altezza dell'albero di decisione associato all'algoritmo SELECTION SORT è:  
 a)  $\Theta(n \log n)$     b)  $\Omega(n!)$     c)  $O(n \log n)$     \*d)  $\Omega(n^2)$
7. A quale delle seguenti classi non appartiene la complessità dell'algoritmo MERGE SORT:  
 a)  $*o(n \log n)$     b)  $\Omega(n)$     c)  $O(n^2)$     d)  $\Theta(n \log n)$
8. Qual è la complessità temporale dell'algoritmo BUCKET SORT applicato ad un array  $A$  di  $n$  elementi in cui l'elemento massimo è pari a  $n^{12} - 5$ ?  
 a)  $\Theta(n + k)$     b)  $\Theta(n)$     \*c)  $O(n^{12})$     d)  $\Theta(n \log n)$
9. Il numero massimo di heap binari distinti generati dall'insieme di 4 elementi  $A = \{1, 2, 3, 4\}$  è:  
 a) 1    b) 4    c) 2    \*d) 3
10. A quale delle seguenti classi non appartiene la complessità dell'algoritmo HEAP SORT:  
 a)  $O(n \log n)$     b)  $\Omega(n)$     c)  $O(n^2)$     \*d)  $o(n \log n)$
11. Una coda di priorità realizzata con una lista lineare ordinata supporta l'estrazione del massimo in:  
 \*a)  $O(1)$     b)  $\Theta(n)$     c)  $\Theta(\log n)$     d)  $\Omega(\log n)$
12. La ricerca del minimo in un heap binomiale di  $n$  elementi costa:    \*a)  $O(\log n)$     b)  $\Theta(1)$     c)  $O(1)$     d)  $\Theta(n)$
13. In un albero binario di ricerca bilanciato di  $n$  elementi, la *ricerca* di un elemento ha complessità:  
 \*a)  $O(\log n)$     b)  $\Omega(n)$     c)  $\Theta(\log n)$     d)  $\Theta(1)$
14. Una tabella ad accesso diretto utilizzata per rappresentare 1000 elementi indicizzati con una chiave a 4 cifre decimali ha un fattore di carico pari a:  
 \*a)  $\alpha = 0,1$     b)  $\alpha = 1$     c)  $\alpha = 0,01$     d)  $\alpha = 10$
15. In un grafo *completo* con 5 vertici, il sottografo indotto da 3 vertici scelti a caso contiene un numero di archi pari a:  
 a) 10    b) 5    \*c) 3    d) 0
16. L'albero BFS (ovvero ottenuto mediante una *visita in ampiezza*) di un grafo connesso di  $n$  vertici ha altezza:  
 a)  $\Theta(n)$     \*b)  $O(n)$     c)  $O(\log n)$     d) 1
17. Dato un grafo pesato con  $n$  vertici ed  $m = O(n)$  archi, l'algoritmo di Dijkstra realizzato con heap di Fibonacci costa:  
 a)  $\Theta(n^2)$     b)  $\Theta(n + m)$     c)  $O(m)$     \*d)  $O(n \log n)$
18. Dato un grafo diretto aciclico con  $n$  vertici ed  $m = O(n)$  archi, l'algoritmo di ordinamento topologico costa:  
 a)  $\Theta(n^2)$     \*b)  $\Theta(n + m)$     c)  $O(n)$     d) non può essere applicato
19. L'operazione  $Union(A, B)$  di 2 insiemi disgiunti  $A, B$  con alberi *QuickFind* con l'euristica dell'unione pesata costa:  
 \*a)  $\Theta(\min(|A|, |B|))$     b)  $\Theta(\max(|A|, |B|))$     c)  $\Theta(|A|)$     d)  $\Theta(|B|)$
20. Dato un grafo connesso con  $n$  vertici ed  $m$  archi, l'algoritmo di Prim esegue un numero di operazioni di decremento delle chiavi pari a:  
 \*a)  $O(m)$     b)  $\Theta(m)$     c)  $O(n)$     d)  $\Theta(n)$

**Griglia Risposte**

	Domanda																			
Risposta	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
a																				
b																				
c																				
d																				



**ESERCIZIO 2 (5 punti) ( Da svolgere sul retro della pagina! )**

Mostrare l'intera esecuzione, passo per passo, dell'algoritmo di Dijkstra sul seguente grafo, assumendo il nodo  $h$  quale nodo sorgente (ogni figura disegnata deve rappresentare lo stato dei nodi del grafo prima di visitare il prossimo nodo).