



| | | | | |
|----------------------|------------------|------------------|------------------|-------|
| Scrivi i tuoi dati ⇒ | Cognome: | Nome: | Matricola: | PUNTI |
| ESERCIZIO 1 | Risposte Esatte: | Risposte Omesse: | Risposte Errate: | |

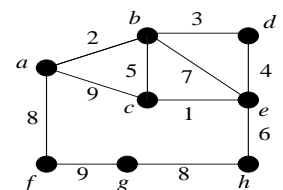
ESERCIZIO 1 (25 punti): Domande a risposta multipla

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 × 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 25. Se tale somma è negativa, verrà assegnato 0.

1. L'algoritmo INSERTION SORT, nel caso migliore costa:
 *a) $O(n)$ b) $\omega(n)$ c) $\Theta(n^2)$ d) $\Omega(n \log n)$
2. Se $f(n) = \Omega(g(n))$ e $g(n) = \omega(h(n))$, allora:
 a) $h(n) = \Omega(f(n))$ b) $f(n) = O(h(n))$ c) $f(n) = \Theta(h(n))$ *d) $f(n) = \omega(h(n))$
3. L'algoritmo SELECTION SORT, nel caso medio costa:
 a) $o(n^2)$ *b) $\Theta(n^2)$ c) $\Theta(n \log n)$ d) $O(n \log n)$
4. L'algoritmo ottimale di fusione di due sequenze ordinate di lunghezza n e n^2 rispettivamente, ha complessità:
 a) $\Theta(n)$ b) $O(n)$ c) $\omega(n^2)$ *d) $\Theta(n^2)$
5. 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) $\Theta(n \log n)$
6. La soluzione dell'equazione di ricorrenza $T(n) = a \cdot T(n/b) + \Theta(n)$, con $T(1) = \Theta(1)$, a, b costanti non negative, $n = b^k$ ed $a > b$ è:
 a) $O(n)$ *b) $O(n^{\log_b a})$ c) $O(n^{\log_a b})$ d) $O(n \log n)$
7. Quale dei seguenti vettori rappresenta un heap binario:
 a) $A=[10,9,6,7,5,11]$ *b) $A=[20,16,9,15,12,8]$ c) $A=[20,16,9,15,17,5,4]$ d) $A=[5,3,2,1,6]$
8. Dato un array non ordinato, la procedura per la costruzione di un heap costa:
 a) $\Theta(n \log n)$ b) $\Theta(1)$ *c) $O(n)$ d) $\Theta(n^2)$
9. Una coda di priorità realizzata con un heap binario supporta l'inserimento di un elemento in:
 *a) $O(\log n)$ b) $\Omega(n)$ c) $\Theta(\log n)$ d) $O(1)$
10. La delimitazione inferiore al problema della ricerca in un insieme di n elementi è:
 a) $\Theta(n \log n)$ *b) $\Omega(n)$ c) $O(\log n)$ d) $\Omega(n \log n)$
11. In un albero binario di ricerca di altezza h , il successore di un elemento può essere determinato in:
 a) $\Theta(\log h)$ b) $O(\log h)$ c) $\Theta(1)$ *d) $O(h)$
12. In un albero AVL di n elementi, l'inserimento di un elemento nel caso migliore induce un numero di rotazioni pari a:
 *a) 0 b) 2 c) $\Theta(\log n)$ d) 1
13. 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) $\alpha = 10$
14. In un grafo *completo* con 4 vertici, il grado di ogni vertice è pari a:
 a) 1 *b) 3 c) 2 d) 4
15. La visita in profondità del grafo  eseguita partendo dal nodo a non può restituire la sequenza di nodi:
 a) $acbdef$ *b) $abcdef$ c) $abdfec$ d) $acefdb$
16. Il nodo a distanza massima da c nel grafo di domanda 15 è:
 a) f *b) a c) d d) b
17. Dato un grafo completo con n vertici rappresentato tramite liste di adiacenza, l'algoritmo di Dijkstra realizzato con heap binario costa:
 a) $\Theta(n^2)$ b) $\Theta(m + n \log n)$ c) $O(n^2)$ *d) $O(n^2 \log n)$
18. L'algoritmo di Floyd e Warshall applicato ad un grafo pesato con un numero di archi $m = \Theta(n)$, ha complessità:
 *a) $\Theta(n^3)$ b) $\Theta(n + m)$ c) $\Theta(n^2 \log n)$ d) $O(m \log n)$
19. Usando gli alberi *QuickUnion* e l'euristica dell'unione pesata *by rank*, il problema della gestione di n insiemi disgiunti sottoposti ad $n - 1$ *Union* ed $m = O(n)$ *Find* può essere risolto in:
 a) $\Theta(n)$ b) $\Theta(m)$ c) $\Theta(m^2)$ *d) $O(n \log n)$
20. In un grafo completo di 10 nodi etichettati da 1 a 10, e tale che l'arco (i, j) ha peso $\min\{i, j\}$, il *minimo albero ricoprente* ha peso:
 a) 10 *b) 9 c) 0 d) 55

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 Prim sul seguente grafo, supponendo di partire dal nodo c.