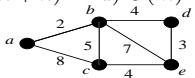
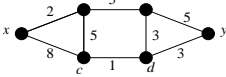




Scrivi i tuoi dati →	Cognome:	Nome:	Matricola:	PUNTI
ESERCIZIO 1	Risposte Esatte:	Risposte Omesse:	Risposte Errate:	

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.

- Un albero binario di ricerca di altezza k contiene:
 - meno di $2^{k+1} - 1$ elementi
 - tra $k + 1$ e $2^{k+1} - 1$ elementi
 - almeno 2^k elementi
 - tra 2^k e $2^{k+1} - 1$ elementi
- In un albero AVL di n elementi, la cancellazione di un elemento nel caso migliore induce un numero di rotazioni pari a:
 - 0
 - 2
 - $\Theta(\log n)$
 - 1
- Dato un albero AVL T contenente n elementi, si consideri l'inserimento di una sequenza di k elementi in T . La nuova altezza di T diventa:
 - $\Theta(n + k)$
 - $\Theta(k + \log n)$
 - $\Theta(\log(n + k))$
 - $\Theta(\log n)$
- In una tavola ad accesso diretto di dimensione m con un fattore di carico $\alpha = 1\%$, l'inserimento di un elemento di un dizionario di n elementi costa:
 - $\Theta(m)$
 - $\Omega(n)$
 - $\Theta(\log n)$
 - $\Theta(1)$
- Siano $h_1(\cdot), h_2(\cdot)$ due funzioni hash. Quale delle seguenti funzioni descrive il metodo di scansione con hashing doppio in una tabella hash di dimensione m per l'inserimento di un elemento con chiave k dopo l' i -esima collisione:
 - $c(k, i) = (h_1(k) + m \cdot h_2(k)) \bmod i$
 - $c(k, i) = (h_1(k) + h_2(k)) \bmod m$
 - $c(k, i) = (h_1(k) + i \cdot h_2(k)) \bmod m$
 - $c(k, i) = (h_1(k) + h_2(k)) \bmod i$
- Siano X e Y due stringhe di lunghezza m ed n . Qual è la complessità dell'algoritmo per la determinazione della distanza tra X e Y basato sulla tecnica della programmazione dinamica?
 - $O(mn)$
 - $O(n)$
 - $O(m + n)$
 - $O(m)$
- La visita in profondità del grafo  eseguita partendo dal nodo d produce un albero DFS di altezza massima:
 - 1
 - 2
 - 4
 - 5
- Un grafo $G = (V, E)$ si dice *bipartito* se l'insieme V può essere partizionato in due sottoinsiemi V_1, V_2 tali che tutti gli archi in E hanno un nodo in V_1 e l'altro in V_2 . Sia dunque $G = (V_1 \cup V_2, E)$ un grafo bipartito tale che $|V_1| = 4, |V_2| = 3$. Quanti archi sono necessari affinché G sia connesso?
 - 3
 - 4
 - 5
 - 6
- In un grafo *non completo* con 6 vertici, il massimo numero di archi contenuti in un suo sottografo indotto da 5 vertici qualsiasi è:
 - 15
 - 5
 - 10
 - 4
- Qual è la *distanza* tra x ed y nel grafo non orientato: 
 - 12
 - 11
 - 3
 - 4
- Si orientino gli archi verticali del grafo della domanda 10 dall'alto verso il basso, e i rimanenti archi da sinistra verso destra. Quali tra i seguenti è un ordinamento topologico dei vertici del grafo:
 - $\langle x, a, c, b, d, y \rangle$
 - $\langle x, c, a, b, d, y \rangle$
 - $\langle x, a, c, d, b, y \rangle$
 - $\langle y, d, b, c, a, x \rangle$
- L'algoritmo di Bellman e Ford applicato ad un grafo pesato con un numero di archi $m = \Theta(n)$, ha complessità:
 - $\Theta(n^2)$
 - $\Theta(n + m)$
 - $\Theta(n^3)$
 - $O(m \log n)$
- Dato un grafo pesato e completo con n vertici rappresentato tramite liste di adiacenza, l'algoritmo di Dijkstra realizzato con un *heap binario* costa:
 - $\Theta(n^2 \log n)$
 - $\Theta(m + n \log n)$
 - $\Theta(n^2)$
 - $O(n \log n)$
- Dato un grafo connesso con n vertici ed m archi rappresentato tramite liste di adiacenza, l'algoritmo di Dijkstra realizzato con una *lista lineare non ordinata* costa:
 - $O(mn)$
 - $\Theta(n + m)$
 - $\Theta(m \log n)$
 - $O(m \log n)$
- Sia d_{xy}^k il costo di un cammino minimo k -vincolato da x a y , secondo la definizione di Floyd e Warshall. Risulta:
 - $d_{xy}^k = \min\{d_{xy}^{k-1}, d_{xv_k}^{k-1} + d_{v_k y}^{k-1}\}$
 - $d_{xy}^k = \min\{d_{xy}^{k-1}, d_{xv_k}^{k-1} + d_{v_k y}^{k-1}\}$
 - $d_{xy}^k = \min\{d_{xy}^{k-1}, d_{xv_k}^k + d_{v_k y}^k\}$
 - $d_{xy}^k = \min\{d_{xy}^k, d_{xv_k}^{k-1} + d_{v_k y}^{k-1}\}$
- L'operazione *Union*(A, B) di 2 insiemi disgiunti A, B con alberi *QuickFind* senza l'euristica dell'unione pesata costa nel caso peggiore:
 - $\Theta(\min(|A|, |B|))$
 - $\Theta(\max(|A|, |B|))$
 - $\Theta(|A|)$
 - $\Theta(|B|)$
- L'operazione *Find*(x) con alberi *QuickUnion* con l'euristica dell'unione pesata *by rank* costa:
 - $\Theta(n)$
 - $\Theta(1)$
 - $\Theta(\log n)$
 - $O(\log n)$
- Dato un grafo pesato con n vertici ed m archi, l'algoritmo di Kruskal esegue un numero di operazioni *UNION*(u, v) pari a:
 - $\Theta(m)$
 - $\Theta(n)$
 - $\Theta(m \log n)$
 - $\Theta(\log n)$
- 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:
 - $O(m)$
 - $\Theta(m)$
 - $O(n)$
 - $\Theta(n)$
- Dato un grafo pesato con n vertici ed m archi, il costo di una fase dell'algoritmo di Borůvka è pari a:
 - $O(m)$
 - $O(n)$
 - $\Theta(m + n \log n)$
 - $\Theta(m \log n)$

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!)

La *somma* di 2 grafi $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$ è un grafo $G = (V, E)$ in cui $V = V_1 \cup V_2$, ed $E = E_1 \cup E_2 \cup \{(x, y) | x \in V_1, y \in V_2\}$. Sia G il grafo ottenuto sommando un ciclo di 4 nodi ed un grafo connesso di 2 nodi. Numerare in modo arbitrario i vertici di G da 1 a 6, e pesare ogni arco come somma dei numeri associati ai vertici incidenti. Restituire quindi il minimo albero ricoprente di G , mostrando l'esecuzione passo per passo dell'algoritmo di Kruskal.