

UNIVERSITÀ DEGLI STUDI DELL'AQUILA
Prova Scritta di **Algoritmi e Strutture Dati con Laboratorio**
Mercoledì 1 Luglio 2020 – 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 $\times$ 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.

- Sia dato in input l'array $A = [3, 5, 2, 1, 8, 4]$; al termine della terza iterazione dell'algoritmo di ordinamento non decrescente INSERTIONSORT2, quale sarà il contenuto dell'array A ?
 a) $A = [1, 2, 3, 4, 5, 8]$ b) $A = [3, 2, 1, 4, 5, 8]$ c) $A = [2, 3, 5, 1, 8, 4]$ *d) $A = [1, 2, 3, 5, 8, 4]$
- Un algoritmo ha una complessità temporale $\Theta(f(n))$ se:
 a) Il tempo di esecuzione $T(n)$ dell'algoritmo su uno specifico input di dimensione n verifica $T(n) = \Theta(f(n))$
 *b) Il tempo di esecuzione $T(n)$ dell'algoritmo su ogni input di dimensione n verifica $T(n) = \Theta(f(n))$
 c) Il tempo di esecuzione medio $T(n)$ dell'algoritmo su un input di dimensione n verifica $T(n) = \Theta(f(n))$
 d) Nel caso migliore, il tempo di esecuzione $T(n)$ dell'algoritmo su un input di dimensione n verifica $T(n) = \Theta(f(n))$
- L'altezza dell'albero di decisione associato al problema dell'ordinamento di n elementi basato su confronti è:
 *a) $\Omega(n \log n)$ b) $\omega(n \log n)$ c) $O(n \log n)$ d) $\Theta(n!)$
- A quale delle seguenti classi di complessità non appartiene la complessità temporale dell'algoritmo QUICKSORT:
 a) $O(n^3)$ b) $\Theta(n^2)$ c) $\Omega(n)$ *d) $o(n^2)$
- Applicando il teorema master, la soluzione dell'equazione di ricorrenza $T(n) = 3 \cdot T(n/3) + n$, è:
 a) $\Theta(n^2)$ b) $\Theta(n^{\log 3})$ *c) $\Theta(n \log n)$ d) $\Theta(n)$
- Sia H_1 un heap binomiale costituito dagli alberi binomiali $\{B_0, B_1, B_2\}$, e sia H_2 un heap binomiale costituito dagli alberi binomiali $\{B_0, B_1, B_3\}$. Da quali alberi binomiali è formato l'heap binomiale ottenuto dalla fusione di H_1 e H_2 ?
 *a) $\{B_1, B_4\}$ b) $\{B_0, B_1, B_2, B_3, B_4\}$ c) $\{B_0, B_0, B_1, B_1, B_2, B_3\}$ d) $\{B_0, B_1, B_2, B_3\}$
- Sia dato un albero AVL di altezza h , e si supponga di cancellare un elemento che provochi lo sbilanciamento dell'albero. Quale tra le seguenti è una terna di valori ammissibili per l'altezza dell'AVL prima della cancellazione, dopo la cancellazione, e infine dopo il ribilanciamento complessivo dell'AVL:
 a) $h, h-1, h-1$ b) $h, h, h-2$ c) $h, h-1, h$ *d) $h, h, h-1$
- Quale tra i seguenti rappresenta lo pseudocodice dell'algoritmo di Bellman&Ford?

a) BELLMAN&FORD(grafo diretto $G = (V, A)$, vertice s) $\rightarrow$ distanze inizializza D con $D_{ss} = 0$ e $D_{sv} = +\infty$ per $v \neq s$; for $1 = 1$ to $n-1$ do for each $(u, v) \in A$ do if $D_{su} + w(u, v) < D_{sv}$ then $D_{sv} \leftarrow D_{su} + w(u, v)$ for each $(u, v) \in A$ do if $D_{su} + w(u, v) < D_{sv}$ then return <i>Errore</i> return D	b) BELLMAN&FORD(grafo diretto $G = (V, A)$, vertice s) $\rightarrow$ distanze inizializza D con $D_{ss} = 0$ e $D_{sv} = +\infty$ per $v \neq s$; for $1 = 1$ to n do for each $(u, v) \in A$ do if $D_{su} + w(u, v) < D_{sv}$ then $D_{sv} \leftarrow D_{su} + w(u, v)$ for each $(u, v) \in A$ do if $D_{su} + w(u, v) < D_{sv}$ then return <i>Errore</i> return D
--	--
- *c) BELLMAN&FORD(grafo diretto $G = (V, A)$, vertice s) $\rightarrow$ distanze
 inizializza D con $D_{ss} = 0$ e $D_{sv} = +\infty$ per $v \neq s$;
 for $i = 1$ to $n-1$ do
 for each $(u, v) \in A$ do
 if $D_{su} + w(u, v) < D_{sv}$ then $D_{sv} \leftarrow D_{su} + w(u, v)$
 for each $(u, v) \in A$ do
 if $D_{su} + w(u, v) < D_{sv}$ then return *Errore*
 return D
- d) BELLMAN&FORD(grafo diretto $G = (V, A)$, vertice s) $\rightarrow$ distanze
 inizializza D con $D_{ss} = 0$ e $D_{sv} = +\infty$ per $v \neq s$;
 for $i = 1$ to $n-1$ do
 for each $(u, v) \in A$ do
 if $D_{su} + w(u, v) < D_{sv}$ then $D_{sv} \leftarrow D_{su} + w(u, v)$
 for each $(u, v) \in A$ do
 if $D_{su} + w(u, v) = D_{sv}$ then return *Errore*
 return D
- Dato un grafo pesato con n vertici ed $m = O(n)$ archi, a quali delle seguenti classi di complessità non appartiene l'algoritmo di Dijkstra realizzato con heap di Fibonacci:
 a) $O(n^2)$ b) $\Theta(m + n \log n)$ *c) $O(m)$ d) $\Theta(m \log n)$
- Quale tra le seguenti è la corretta definizione di ordinamento topologico di un grafo diretto $G = (V, A)$?
 a) è una funzione iniettiva $\sigma : V \mapsto \{1, \dots, n\}$ tale che se $\exists$ un cammino da u a v in G , con $u \neq v$, allora $\sigma(u) < \sigma(v)$.
 *b) è una funzione biettiva $\sigma : V \mapsto \{1, \dots, n\}$ tale che se $\exists$ un cammino da u a v in G , con $u \neq v$, allora $\sigma(u) < \sigma(v)$.
 c) è una funzione biettiva $\sigma : V \mapsto \{1, \dots, n\}$ tale che $\sigma(u) < \sigma(v)$ se e solo se $\exists$ un cammino da u a v in G , con $u \neq v$.
 d) è una funzione iniettiva $\sigma : V \mapsto \{1, \dots, n\}$ tale che $\sigma(u) < \sigma(v)$ se e solo se $\exists$ un cammino da u a v in G , con $u \neq v$.

Griglia Risposte

[illegible]