

Prova Scritta di **Algoritmi e Strutture Dati con Laboratorio**
Mercoledì 7 settembre 2022 – Prof. Guido Proietti (Modulo di Teoria)

Scrivi i tuoi dati \Rightarrow	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.

1. La dimensione dell'input degli algoritmi studiati per calcolare l' n -esimo numero della sequenza di Fibonacci è $k = \Theta(\log n)$. Qual è la complessità temporale in funzione della dimensione dell'input dell'algoritmo ricorsivo basato sul prodotto tra matrici?
a) $\omega(k)$ b) $\Theta(n)$ *c) $\Theta(k)$ d) $\Theta(k \log k)$
2. Sia dato in input l'array $A = [n, n-1, \dots, 3, 2, 1]$, e si supponga di applicare su di esso gli algoritmi di ordinamento non decrescente SELECTIONSORT, INSERTIONSORT1 e INSERTIONSORT2. Quale dei tre algoritmi esegue il minor numero di operazioni asintoticamente?
a) SELECTIONSORT b) INSERTIONSORT1 c) INSERTIONSORT2 *d) Sono tutt'e tre equivalenti
3. L'algoritmo ottimale di fusione di due sequenze ordinate di lunghezza p e q rispettivamente, ha complessità:
a) $\Theta(p \cdot q)$ b) $\Theta(p)$ c) $\omega(p+q)$ *d) $\Theta(p+q)$
4. 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 $O(n)$
5. Sia H un heap binomiale costituito dagli alberi binomiali $\{B_0, B_1, B_5\}$, e si supponga di eseguire su di esso un'operazione di *deleteMin()*. Da quali alberi binomiali è formato l'heap binomiale risultante?
*a) $\{B_1, B_5\}$ b) $\{B_0, B_1, B_5\}$ c) $\{B_0, B_1, B_2, B_3, B_4\}$ d) $\{B_0, B_5\}$
6. Dato un albero binario di ricerca di n elementi, la cancellazione di un elemento restituisce un albero avente al massimo altezza:
*a) $n-2$ b) n c) $\Theta(\log n)$ d) $n-1$
7. Si supponga di inserire la sequenza di chiavi 10, 22, 31 (in quest'ordine) in una tavola hash di lunghezza $m = 3$ (ovvero con indici 0, 1, 2) utilizzando l'indirizzamento aperto con funzione hash $h(k) = k \bmod 3$, e risolvendo le collisioni con il metodo della scansione lineare. Quale sarà la tavola hash finale?
a) $A = [10, 22, 31]$ b) $A = [22, 31, 10]$ c) $A = [31, 22, 10]$ *d) $A = [31, 10, 22]$
8. Sia dato un grafo non diretto G con n vertici, numerati da 1 ad n , ed $n-1$ archi, disposti in modo arbitrario, ma in modo tale da garantire la connessione. Si orientino ora gli archi in modo arbitrario, e si applichi l'algoritmo di ordinamento topologico rispetto al nodo sorgente etichettato 1. La complessità risultante è pari a:
a) $\Theta(n^2)$ *b) $\Theta(n)$ c) $\Theta(n \log n)$ d) indefinita (non è detto che l'algoritmo possa essere applicato)
9. Dato un grafo diretto senza cicli di costo negativo, e preso un nodo arbitrario s del grafo, si applichi su di esso l'algoritmo di Bellman&Ford per trovare l'albero dei cammini minimi radicato in s . Alla fine dell' i -esima iterazione, l'algoritmo avrà trovato (una sola delle seguenti affermazioni è sempre vera):
a) Tutti i cammini minimi da s verso gli i nodi più vicini ad s
b) Al più i cammini minimi da s
c) Almeno un cammino minimo costituito da esattamente i archi
*d) Tutti i cammini minimi da s costituiti da al più i archi
10. Dato un grafo completo con n vertici, l'algoritmo di Prim eseguito con un *array lineare non ordinato* costa:
a) $o(n^2)$ b) $\omega(n^2)$ c) $\Theta(n^2 \log n)$ *d) $\Theta(n^2)$

Griglia Risposte

[illegible]