

Università degli Studi dell'Aquila

Prova di Recupero di Algoritmi e Strutture Dati con Laboratorio Mercoledì 11 Luglio 2012 - Proff. Guido Proietti e Giovanna Melideo

Scrivi i tuoi dati \Longrightarrow	Cognome:	Nome:	Matricola:

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

- 1. Un problema ha una delimitazione inferiore alla complessità temporale $\Omega(f(n))$ se:
 - *a) Tutti gli algoritmi per la sua risoluzione hanno una delimitazione inferiore alla complessità computazionale pari a $\Omega(f(n))$
 - b) Tutti gli algoritmi per la sua risoluzione hanno una delimitazione superiore alla complessità computazionale pari a O(f(n))
 - c) Esiste un algoritmo per la sua risoluzione che ha una delimitazione inferiore alla complessità computazionale pari a $\Omega(f(n))$
 - d) Esiste un algoritmo per la sua risoluzione che ha una delimitazione superiore alla complessità computazionale pari a O(f(n))
- 2. Sia f(n) il costo dell'algoritmo Insertion Sort nel caso migliore, e sia g(n) il costo dell'algoritmo Merge Sort nel caso medio. Quale delle seguenti relazioni asintotiche è vera:

d) $f(n) = \Omega(g(n))$ *a) f(n) = o(g(n)) b) $f(n) = \Theta(g(n))$ c) $f(n) = \omega(g(n))$

3. Quale tra i seguenti rappresenta lo pseudocodice dell'algoritmo HEAPSORT:

 $^{'}$ Heapsort(A) Heapify(A)heapsize[A] = nfor (i = n) down to 1 do scambia A[1] con A[i]heapsize[A] = heapsize[A] - 1Fixheap(1, A)

Heapsort(A)Heapify(A)heapsize[A] = nfor (i = n) down to 2 do scambia A[n] con A[i]heapsize[A] = heapsize[A] - 1Fixheap(1, A)

HEAPSORT(A)Heapify(A)heapsize[A] = nfor (i = n) down to 2 do scàmbia A[1] con A[i]heapsize[A] = heapsize[A] - 1Fixheap(1, A)

 $H_{EAPSORT}(A)$ Heapify(A)heapsize[A] = nfor (i = n) down to 2 do scambia A[1] con A[i]heapsize[A] = heapsize[A] - 1Fixheap(n, A)

- 4. Qual è la complessità spaziale dell'algoritmo Integer Sort applicato ad un array A di n elementi in cui $A[i] = 3i^4 + i$ per i = 1, ..., n? b) $\Theta(n)$ a) $\Theta(n^5)$ *c) $\Theta(n^4)$ d) $\Theta(n \log n)$
- 5. 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) $\Omega(\log n)$
- 6. In un heap binomiale di n elementi, l'operazione di decremento della chiave di un elemento ha complessità: *d) $O(\log n)$ a) $\Theta(1)$ b) O(1)c) $\Theta(n)$
- 7. Si consideri il semplice albero AVL di 5 elementi, nella cui radice compare la chiave 5, nel figlio sinistro della radice compare la chiave 2, nel figlio destro della radice compare la chiave 4, nel figlio sinistro del figlio sinistro della radice compare la chiave 1, e infine nel figlio destro del figlio sinistro della radice compare la chiave 3. Si supponga ora di cancellare l'elemento nella radice dell'AVL. Quale albero AVL risulterà?
 - a) Nella radice compare la chiave 3, nel figlio sinistro della radice compare la chiave 1, nel figlio destro della radice compare la chiave 4, e nel figlio destro del figlio sinistro della radice compare la chiave 2.
 - *b) Nella radice compare la chiave 3, nel figlio sinistro della radice compare la chiave 2, nel figlio destro della radice compare la chiave 4, e nel figlio sinistro del
 figlio sinistro della radice compare la chiave ${\bf 1}.$
 - c) Nella radice compare la chiave 2, nel figlio sinistro della radice compare la chiave 1, nel figlio destro della radice compare la chiave 3, e nel figlio destro del figlio destro della radice compare la chiave 4.
 - d) Nella radice compare la chiave 2, nel figlio sinistro della radice compare la chiave 1, nel figlio destro della radice compare la chiave 4, e nel figlio sinistro del figlio destro della radice compare la chiave 3.
- 8. 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:

a) $c(k, i) = (h_1(k) + m \cdot h_2(k)) \mod i$ b) $c(k, i) = (h_1(k) + h_2(k)) \mod m$ *c) $c(k, i) = (h_1(k) + i \cdot h_2(k)) \mod m$ d) $c(k, i) = (h_1(k) + h_2(k)) \mod i$

9. 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) ind

d) indefinita (non è detto che l'algoritmo possa essere applicato)

10. Dato un grafo pesato con n vertici ed m = O(n) archi, l'algoritmo di Prim realizzato con heap di Fibonacci costa: *d) $O(n \log n)$ a) $\Theta(n^2)$ b) $\Theta(n+m)$ c) O(m)

Griglia Risposte

	Domanda									
Risposta	1	2	3	4	5	6	7	8	9	10
a										
b										
С										
d										