

Università degli Studi dell'Aquila

Prova di Recupero di **Algoritmi e Strutture Dati con Laboratorio** Martedì 19 Febbraio 2013 – Proff. Guido Proietti e Giovanna Melideo

Scrivi i tuoi dati ⇒	Cagnama	Nome:	Matricola:
Scrivi i tuoi dati —>	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 \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. Quale delle seguenti relazioni di ricorrenza descrive la complessità dell'algoritmo Fibonacci2? a) T(n) = 2T(n/2) + O(1) se $n \ge 2$, T(1) = O(1) se n = 1 b) T(n) = 2T(n/4) + O(1) se $n \ge 2$, T(1) = O(1) se n = 1 *c) T(n) = 2 + T(n-1) + T(n-2) se $n \ge 3$, T(1) = T(2) = 1 se n = 1 d) T(n) = 2 + T(n-1) se $n \ge 2$, T(1) = 1 se n = 1
- 2. Siano f(n) e g(n) i costi dell'algoritmo Insertion Sort nel caso migliore e Quicksort in quello medio, rispettivamente. Quale delle seguenti relazioni asintotiche è vera:
 - *a) f(n) = o(g(n)) b) $f(n) = \Theta(g(n))$ c) $f(n) = \omega(g(n))$ d) $f(n) = \Omega(g(n))$
- 3. A quale delle seguenti classi asintotiche non appartiene la complessità dell'algoritmo MERGE SORT: *a) $o(n \log n)$ b) $\omega(n)$ c) $O(n \log n)$ d) $o(n^2)$
- 4. La procedura Heap-Insert(A, 4) applicata al vettore A = [12, 9, 3, 6, 5, 2, 1, nil] rappresentante un heap binario restituisce: a) A = [10, 9, 4, 6, 5, 3, 2, 1] b) A = [12, 9, 4, 6, 5, 3, 2, 1] c) A = [12, 9, 4, 6, 5, 2, 3, 1] *d) A = [12, 9, 3, 6, 5, 2, 1, 4]
- 5. Si supponga di inserire la sequenza di chiavi 23,11,3 (in quest'ordine) in una tavola hash di lunghezza m=3 (ovvero con indici 0,1,2) utilizzando l'indirizzamento aperto con funzione hash $h(k)=k \mod 3$, e risolvendo le collisioni con il metodo della scansione quadratica con $c_1=c_2=1$. Quale sarà la tavola hash finale?
 - a) A = [11, 3, 23] b) A = [23, 3, 11] c) A = [3, 23, 11] *d) A = [3, 11, 23]
- 6. Si consideri il grafo G = (V, E) con $V = \{1, 2, 3, 4, 5\}$ ed $E = \{(1, 2), (1, 5), (2, 3), (2, 4)\}$. Quali delle seguenti affermazioni è <u>falsa</u>:
 a) G è bipartito
 b) se radichiamo G in 1 e orientiamo tutti gli archi dalla radice verso le foglie, otteniamo un albero binario quasi completo
 c) il diametro di G, ovvero la distanza massima tra due nodi in G, è pari a 4 d) G ha grado 3
- 7. Dato un grafo connesso di n nodi ed m archi, per quale valore (asintotico) di m si ha che l'implementazione dell'algoritmo di Dijkstra con heap binari ha la stessa complessità temporale dell'implementazione con array non ordinati?

 *a) $m = \Theta(n^2/\log n)$ b) $m = \Theta(n^2)$ c) $m = \Theta(n)$ d) mai
- 8. Dato un grafo connesso di n nodi ed m archi, si supponga di voler determinare tutte le distanze da un nodo sorgente fissato. Per quale valore (asintotico) di m si ha che l'implementazione dell'algoritmo di Dijkstra con liste ordinate ha la stessa complessità temporale dell'implementazione di Floyd e Warshall?
 - *a) $m = \Theta(n^2)$ b) $m = \Theta(n)$ c) per ogni valore di m d) mai
- 9. In un grafo completo di 10 nodi etichettati da 1 a 10, e tale che l'arco (i, j) ha peso |i j|, il minimo albero ricoprente ha peso: a) 10 **b) 9 **c) 0 **d) 55
- 10. Dato un grafo connesso di *n* nodi ed *m* archi, per quale valore (asintotico) di *m* si ha che l'implementazione di Prim con liste non ordinate è strettamente più efficiente dell'implementazione di Kruskal con alberi QuickUnion con euristica di bilanciamento *union*
 - a) $m = o(n \log n)$ b) sempre c) mai *d) $m = \omega(n^2/\log n)$

Griglia Risposte

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