Università degli Studi dell'Aquila

Prova Scritta di **Algoritmi e Strutture Dati con Laboratorio**

Martedì 16 Gennaio 2018 – Prof. Guido Proietti (Modulo di Teoria)

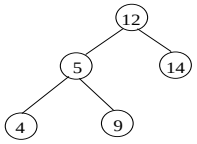
Scrivi i tuoi dati ⟹	Cognome:	Nome:	Matricola:	PUNTI
ESERCIZIO 1	Risposte Esatte:	Risposte Omesse:	Risposte Errate:	

ESERCIZIO 1 (Domande a risposta multipla): 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 e normalizzando su base 30. Se tale somma è negativa, verrà assegnato 0.

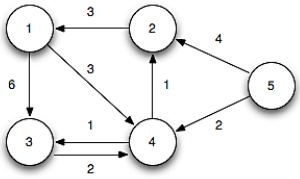
1. Dato un problema P, un algoritmo per la sua risoluzione $\mathcal{A}(P)$ si dice *ottimale* se e solo se:
   a) l'upper bound di P coincide asintoticamente con il costo nel caso peggiore di $\mathcal{A}(P)$;
   b) l'upper bound di P coincide asintoticamente con il costo nel caso migliore di $\mathcal{A}(P)$;
   c) il lower bound di P coincide asintoticamente con il costo nel caso migliore di $\mathcal{A}(P)$;
   *d) il lower bound di P coincide asintoticamente con il costo nel caso peggiore di $\mathcal{A}(P)$.
2. Siano $f(n)$ e $g(n)$ i costi dell'algoritmo Selection Sort e Quicksort nel caso 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) $f(n) = O(g(n))$
3. A quale algoritmo di ordinamento corrisponde il seguente pseudocodice (il primo elemento dell'array è indicizzato $A[1]$):

```
Sort(A)
  for (k = 1) to n − 2 do
    m = k + 1
    for (j = k + 2) to n do
      if (A[j] < A[m]) then m = j
    scambia A[m] con A[k + 1]
```

   a) Selection Sort b) Insertion Sort c) Quicksort *d) L'algoritmo non ordina correttamente.
4. Siano date $\Theta(\log n)$ sequenze di numeri interi arbitrari, ciascuna di dimensione $\Theta(n)$. Si supponga di fondere tra di loro tali sequenze, ottenendo quindi un'unica sequenza, e di applicare su di essa l'algoritmo di ordinamento Merge Sort. Qual è la complessità computazionale risultante?
   a) $\Theta(n \log n)$ b) $\Theta(n^2)$ c) $\Theta(n \log n \log \log n)$ *d) $\Theta(n \log^2 n)$
5. Come si esegue l'operazione `increaseKey`(elem e, chiave Δ) di un elemento con chiave k in un heap binomiale?
   a) eseguendo `insert`$(e, k + \Delta)$ b) eseguendo `decreaseKey`$(e, -\Delta)$
   c) eseguendo `delete`(e) seguita da `insert`(e, Δ) *d) eseguendo `delete`(e) seguita da `insert`$(e, k + \Delta)$
6. Dato l'albero AVL in figura si supponga di inserire l'elemento con chiave 8. La visita in ordine *posticipato* dell'AVL ribilanciato restituisce:
   *a) 4,8,5,14,12,9 b) 4,8,14,5,12,9 c) 9,5,12,4,8,14 d) 9,5,4,8,12,14



7. Dato il grafo G in figura si applichi l'algoritmo di Bellman&Ford con sorgente il nodo 1, esaminando gli archi in ordine lessicografico. Qual è la stima di distanza dal nodo 1 al nodo 3 alla fine della prima iterazione?



   a) $+\infty$ *b) 4 c) 6 d) Non si può applicare l'algoritmo di Bellman&Ford
8. Dato un grafo pesato $G = (V, E)$ con n vertici ed m archi, e presi 2 vertici u, v tali che l'arco $(u, v) \in E$, quanto costa trovare il cammino minimo tra u e v applicando l'algoritmo di Dijkstra realizzato con una *heap binario*?
   a) $\Theta(1)$ b) $\Theta(m)$ c) $O(n^2)$ *d) $O(m \log n)$
9. Si consideri il grafo di Domanda 7, e si applichi l'algoritmo di Floyd&Warshall, con la numerazione dei vertici data in figura. Qual è la lunghezza del cammino minimo 3-vincolato tra il nodo 3 e il nodo 1?
   *a) $+\infty$ b) 3 c) 6 d) Non si può applicare l'algoritmo di Floyd&Warshall
10. Dato un grafo pesato con n vertici e $m = \Theta(n)$ archi, quale delle seguenti classi caratterizza meglio la complessità dell'algoritmo di Kruskal eseguito con alberi QuickUnion con euristica di bilanciamento *union by size*?
    a) $\Theta(n)$ *b) $\Theta(n \log n)$ c) $\Theta(n^2)$ d) $O(n^2)$

Griglia Risposte

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