Algoritmi e Strutture Dati

a.a. 2009/10

Compito del 19/2/2010

Cognome: _____ Nome: _____

Matricola: _____ E-mail: _____

Parte I

(30 minuti; ogni esercizio vale 2 punti)

1. Completare la seguente tabella indicando la complessità delle operazioni che si riferiscono a un dizionario di *n* elementi. Si noti che tutte le operazioni, tranne la **Ricerca**, assumono di aver già raggiunto l'elemento *x* a cui si applica l'operazione.

	Minimo	**Ricerca**	**Cancellazione**	**Successore**	**Costruzione**
Lista semplice ordinata					
minHeap					

2. Risolvere le seguenti relazioni di ricorrenza (giustificando le risposte):

   (a) $T(n) = 2 \cdot T(n/2) + \sqrt{n}$ *(b)* $T(n) = 27 \cdot T(n/3) + n^3$ *(c)* $T(n) = 5 \cdot T(n/4) + n^2$

3. Si definiscano le classi di complessità P, NP, NPC, e si dica quale delle seguenti affermazioni è vera, quale è falsa e quale è improbabile (giustificando le risposte):

   (a) $NP \subseteq P$ (b) $P \subseteq NP$ (c) P u NPC = NP (d) $P \cap NPC \neq \varnothing \Rightarrow P = NP$

Algoritmi e Strutture Dati

a.a. 2009/10

Compito del 19/2/2010

Cognome: _____ Nome: _____

Matricola: _____ E-mail: _____

Parte II

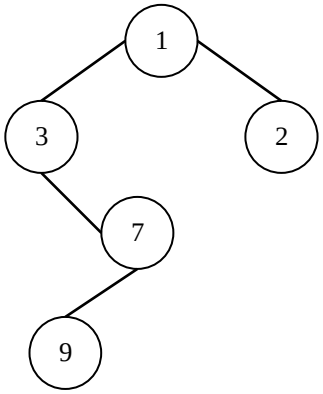
(2.5 ore; ogni esercizio vale 6 punti)

1. Un albero binario si dice **t-bilanciato** se per ogni suo nodo vale la proprietà: le altezze dei sottoalberi radicati nei suoi due figli differiscono per al più **t** unità.
   a. Dato un albero binario, scrivere una funzione **efficiente** in C che restituisca il minimo valore **t** per cui l'albero risulti **t-bilanciato**.
   b. Discutere la complessità della soluzione trovata.

   Si deve utilizzare il seguente tipo per la rappresentazione di un albero binario:

```
typedef struct node{
    int key;
    struct node * left;
    struct node * right;
} * Node;
```

2. Dato il seguente albero:



   a. Eseguire una visita in ordine anticipato e una visita in ordine simmetrico elencando nei due casi la sequenza dei nodi incontrati.
   b. Progettare un algoritmo che dati due vettori contenenti rispettivamente i valori dei nodi tutti **distinti** ottenuti da una visita in ordine anticipato e da una visita in ordine simmetrico di un albero binario, ricostruisca l'albero binario.
      Un possibile prototipo della funzione è:

      ricostruisci(array vant, int infant, int supant, array vsim, int infsim, int supsim) →Node

      Suggerimento : il vettore vant permette di determinare la radice dell'albero mentre il vettore vsim la suddivisione dei nodi fra sottoalbero sinistro e destro.
   c. Analizzare la complessità dell'algoritmo nel caso pessimo.

La rappresentazione dell'albero binario utilizza esclusivamente i campi **left, right** e **key.**

3. La seguente tabella fornisce le distanze (in unità di 100 miglia) tra gli aeroporti delle città di Londra, Città del Messico, New York, Parigi, Pechino e Tokyo:

	L	*CM*	*NY*	*Pa*	*Pe*	*T*
L	–	56	35	2	51	60
CM	56	–	21	57	78	70
NY	35	21	–	36	68	68
Pa	2	57	36	–	51	61
Pe	51	78	68	51	–	13
T	60	70	68	61	13	–

Utilizzando sia l'algoritmo di Kruskal che quello di Prim, si determini un albero di copertura minimo per il grafo corrispondente. Le esecuzioni di entrambi gli algoritmi dovranno essere simulate accuratamente.

4. Si scriva l'algoritmo di Floyd-Warshall per il problema dei cammini minimi tra tutte le coppie, si fornisca la sua complessità computazionale, si dimostri la sua correttezza e si simuli accuratamente la sua esecuzione sulla seguente matrice:

$$W = \begin{bmatrix} 0 & 2 & 4 & \infty & 3 \\ 2 & 0 & 8 & \infty & 1 \\ 6 & 2 & 0 & 4 & 3 \\ 1 & \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & 1 & 0 \end{bmatrix}.$$