Algoritmi e Strutture Dati
a.a. 2013/14

Compito del 29/05/2014

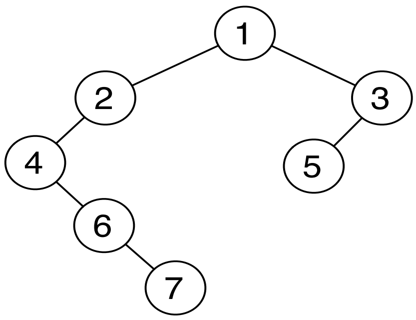
Cognome: ______________________________ Nome: ______________________________

Matricola: ______________________________ E-mail: ______________________________

Parte I

(30 minuti; ogni esercizio vale 2 punti)

1. Dato il seguente albero



   Eseguire una visita in preordine, una visita in ordine simmetrico e una visita in postordine elencando nei tre casi la sequenza dei nodi incontrati.

2. Si determini la complessità asintotica dell'algoritmo **AlgA** definito come segue:

```
AlgA(n)
s ← 0
    for i ← 1 to n
              do s ← s + AlgB(n)
return s

AlgB(m)
if m = 1
    then return 0
    else return B(m/2) + m
```

3. Si scriva un algoritmo di complessità $O(n^2 \log n)$ per determinare le distanze tra tutte le coppie di vertici in un grafo sparso G avente pesi sugli archi positivi, dove n è il numero di vertici in G.

Algoritmi e Strutture Dati
a.a. 2013/14

Compito del 29/05/2014

Cognome: ______________________________ Nome: ____________________________________

Matricola: ______________________________ E-mail: ___________________________________

Parte II

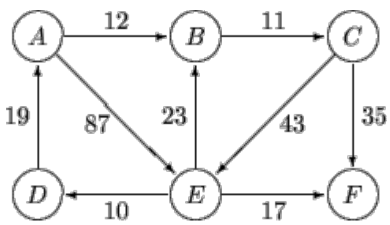
(2.5 ore; ogni esercizio vale 6 punti)

1. L'operazione *Heap-Delete(A, i)*, cancella l'elemento nel nodo *i* dall'heap *A*. Implementare la procedura *Heap-Delete* in modo che il suo tempo di esecuzione sia *O(lg n)* per un max-heap di *n* elementi.

   Per l'esame da **12 CFU**, deve essere fornita **una funzione C**.
   Per l'esame da **9 CFU**, è sufficiente specificare lo pseudocodice.

2. Progettare un algoritmo **efficiente** di tipo **divide et impera** che dato un vettore di interi restituisce *true* se tutti i valori sono **distinti**, *false* altrimenti. Analizzare la complessità dell'algoritmo proposto.

3. Si scriva l'algoritmo di Bellman-Ford, si dimostri la sua correttezza, si fornisca la sua complessità computazionale e si simuli accuratamente la sua esecuzione sul seguente grafo (utilizzando il vertice A come sorgente):



4. Sia $G = (V, E)$ un grafo pesato non orientato e connesso e sia $w : E \rightarrow R$ la funzione peso. Sia $T_{\min}$ un albero di copertura minimo di G e sia T un qualsiasi altro albero di copertura (non necessariamente minimo). Inoltre sia $(u,v) \in E$ un arco di peso massimo in $T_{\min}$ e $(x,y) \in E$ un arco di peso massimo in T. Si dimostri che:

$$w(u,v) \leq w(x,y).$$

   In altri termini, fra tutti gli alberi di copertura, l'albero di copertura minimo ha il più piccolo arco di peso massimo. (Suggerimemto: si ragioni per assurdo e si usi la classica tecnica del "taglia-e-incolla".)