Algoritmi e Strutture Dati
a.a. 2011/12

Compito del 26/1/2012

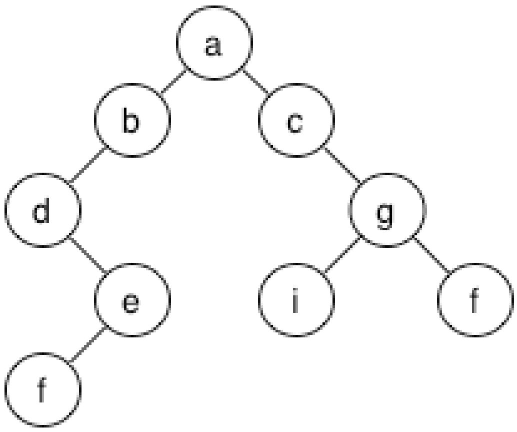
Cognome: ______________________ Nome: ______________________

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 diano le definizioni di O, Ω, Θ, o, ω e si stabilisca, utilizzando le definizioni date, se $7n^3 + 2n = \Theta(n^3)$.

3. Si definiscano le classi di complessità P, NP, NPC, e si mostri che se esiste un problema NP-completo risolvibile polinomialmente allora P = NP.

Algoritmi e Strutture Dati

a.a. 2011/12

Compito del 26/1/2012

Cognome: ______________________________ Nome: ______________________________

Matricola: ______________________________ E-mail: ______________________________

Parte II

(2.5 ore; ogni esercizio vale 6 punti)

1. Dato un albero binario con chiavi intere positive e negative, progettare un algoritmo **efficiente** che restituisca la somma totale di tutte le chiavi dell'albero.
   Definire un secondo algoritmo, anch'esso **efficiente**, che restituisca il massimo peso di tutti i sottoalberi, dove il peso di un sottoalbero è la somma di tutte le chiavi dei suoi nodi.

   Valutare la complessità dei due algoritmi.

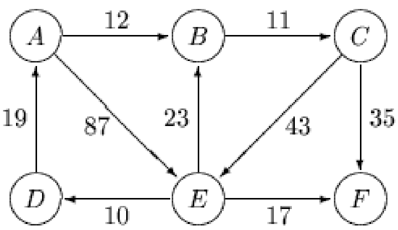
   La rappresentazione dell'albero binario utilizza esclusivamente i campi **left, right** e **key.**

2. Modificare la funzione PARTITION del quicksort per ottenere una funzione PARTITION'(A, p, r) che **permuta** gli elementi di A[p..r] e restituisce due indici *q* e *t*, con $p \leq q \leq t \leq r$, tali che
   a. tutti gli elementi di A[q..t] siano uguali
   b. ogni elemento di A[p..q-1] sia minore di A[q]
   c. ogni elemento di A[t+1..r] sia maggiore di A[q]

   Come PARTITION, anche PARTITION' dovrà richiedere un tempo Θ(r-p).

   Dimostrare la correttezza di PARTITION' definendo un'invariante di ciclo.

3. Si scriva l'algoritmo di Dijkstra, 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 non orientato, connesso e pesato avente tutti i pesi distinti. Sia C un ciclo di G e sia *(u,v)* l'arco in C avente peso massimo. Si stabilisca se esiste o meno in G un albero di copertura minimo che contiene l'arco *(u,v)*. Potremmo dire lo stesso se i pesi sugli archi non fossero distinti? Giustificare formalmente le risposte.