

# Algoritmi e Strutture Dati

a.a. 2009/10

Compito del 14/9/2010

Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_

Matricola: \_\_\_\_\_ E-mail: \_\_\_\_\_

## Parte I

*(30 minuti; ogni esercizio vale 2 punti)*

1. Determinare il numero di **nodi interni** in un albero **d-ario completo** in funzione dell'altezza  $h$  e dimostrare per induzione la correttezza della risposta.
2. Si diano le definizioni di  $O$ ,  $\Omega$  e  $\Theta$  e si stabilisca, utilizzando le definizioni date, se  $3n^2 + 7n = \Theta(n^2)$ .
3. Si definisca la relazione di "riducibilità polinomiale" tra problemi ( $\leq_p$ ) e si stabilisca se valgono le seguenti proprietà: *a)* riflessiva, *b)* simmetrica, *c)* transitiva (giustificando tecnicamente le risposte).

# Algoritmi e Strutture Dati

a.a. 2009/10

Compito del 14/9/2010

Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_

Matricola: \_\_\_\_\_ E-mail: \_\_\_\_\_

## Parte II

(2.5 ore; ogni esercizio vale 6 punti)

1. Dato un albero binario  $T$ , definire una funzione **efficiente** in C che restituisca una copia  $T'$  di  $T$ , che contenga anche, in ogni nodo, il numero di nodi del sottoalbero di cui è radice (radice inclusa).

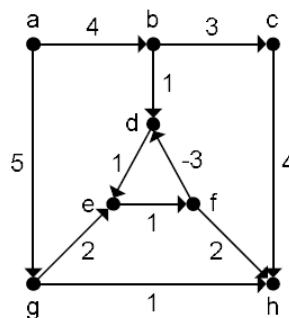
Discutere la complessità della soluzione trovata.

Il tipo dell'albero  $T$  è:

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

Modificare in modo adeguato il tipo dell'albero  $T$  per ottenere il tipo dell'albero  $T'$ .

2. Progettare un algoritmo che, ricevuto in input un intero  $k$  e un array  $a$ , **non ordinato**, di  $n$  elementi **distinti**, restituisca il  $k$ -esimo elemento più piccolo di  $a$ . La soluzione **deve** essere di costo in tempo  $O(n \log k)$  e utilizza uno heap di  $k$  elementi.
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. Si scriva l'algoritmo di Dijkstra, si dimostri la sua correttezza e si fornisca la sua complessità computazionale. Si supponga inoltre di cambiare l'istruzione **while**  $Q \neq \emptyset$  dell'algoritmo, con la seguente: **while**  $|Q| > 1$ . Questa variazione fa eseguire il ciclo  $|V| - 1$  volte invece di  $|V|$ . L'algoritmo proposto è corretto? (Giustificare "tecnicamente" la risposta).