

Algoritmi e Strutture Dati
a.a. 2011/12

Compito del 10/09/2012

Cognome: _____

Nome: _____

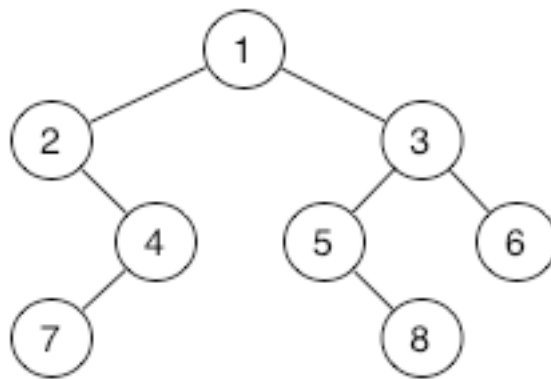
Matricola: _____

E-mail: _____

Parte I

(30 minuti; ogni esercizio vale 2 punti)

1. Dato il seguente albero



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

2. Sia G un grafo orientato sparso con pesi sugli archi positivi. Si scriva un algoritmo per determinare le distanze tra tutte le coppie di vertici in G che sia asintoticamente più efficiente dell'algoritmo di Floyd-Warshall. (Si giustifichi tecnicamente la risposta.)
3. Si stabilisca se il seguente *ragionamento* è errato o meno, giustificando la risposta:
- Il problema ISOMORFISMO-DI-GRAFI, che consiste nel determinare se due grafi sono identici (isomorfi), può essere ricondotto al problema CLIQUE su un grafo ausiliario detto "grafo di associazione";
 - Il problema CLIQUE è NP-completo;
 - Quindi, il problema ISOMORFISMO-DI-GRAFI è NP-completo.

Algoritmi e Strutture Dati

a.a. 2011/12

Compito del 10/09/2012

Cognome: _____

Nome: _____

Matricola: _____

E-mail: _____

Parte II

(2.5 ore; ogni esercizio vale 6 punti)

1. Dato un array v **ordinato in senso crescente** di n interi, il cui valore può essere solo 0 e 1, progettare un algoritmo **efficiente**, di tipo *divide-et-impera*, che restituisca il numero di occorrenze del numero 1 in v .

Calcolare la complessità al caso peggio dell'algoritmo proposto indicando, e risolvendo, la corrispondente relazione di ricorrenza.

2. Dato un albero di ricerca T , progettare un algoritmo **efficiente** che restituisca il numero massimo di ripetizioni di una chiave in T e analizzarne la complessità.

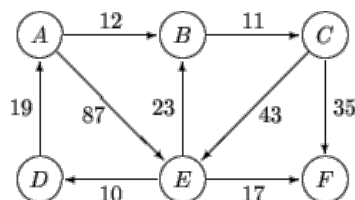
Non si possono usare strutture ausiliarie di dimensione $O(n)$ dove n è il numero dei nodi dell'albero.

Devono essere definite esplicitamente eventuali funzioni/procedure ausiliarie. Si consideri la rappresentazione dell'albero binario di ricerca che utilizza i campi **left**, **right**, **p** e **key**.

3. 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? Perché?

Infine, si simuli accuratamente l'esecuzione dell'algoritmo (modificato) sul seguente grafo, utilizzando sia il vertice A che il vertice F come sorgente:



4. Si enunci e si dimostri il teorema fondamentale delle ricorrenze e lo si utilizzi per risolvere le seguenti ricorrenze (spiegando in quali casi del teorema ricade la soluzione di ciascuna equazione):

a) $T(n) = 4T(n/2) + \log n$

b) $T(n) = 2T(n/2) + n^3$

c) $T(n) = 7T(n/2) + n^2$