

Università Degli Studi dell'Aquila

V appello di Laboratorio di Algoritmi e Strutture Dati - A.A. 2022/2023

Giovedì 13 luglio 2023 – Dott.ssa Giovanna Melideo (Durata: 1:30 h)

Svolgere i seguenti esercizi avendo come riferimento il linguaggio JAVA.

ESERCIZIO 1 Un corso di Studi desidera gestire automaticamente l'assegnazione delle tesi ai propri studenti. Ogni studente è univocamente individuato dalla sua matricola. Sviluppare le seguenti due classi:

1. Una classe **Tesi**, avente come variabili d'istanza il titolo, il nome ed il cognome del docente proponente (si considerino uguali due tesi aventi lo stesso titolo);
2. Una classe **Assegnazioni** che gestisce le assegnazioni delle tesi ai laureandi:

```
public class Assegnazioni {

    private ArrayList<Tesi> tesi_da_assegnare;
    //lista di tesi proposte dai docenti e non ancora assegnate, senza duplicazioni

    private TreeMap<String, Tesi> assegnazioni;
    // mappa degli studenti (individuati dalla matricola di tipo String) con tesi assegnata

    public Assegnazioni() {...}
    // Il costruttore inizializza un oggetto Assegnazioni con lista e mappa vuote.

    public boolean aggiungiAssegnazione(String matricola, Tesi tesi){...}
    /* Se la tesi è già presente nella lista di tesi da assegnare, assegnarla allo studente
    specificato dalla matricola ed eliminarla contestualmente dalla lista. Se la tesi non è
    presente nella lista di tesi da assegnare, può essere assegnata allo studente (nuovo
    assegnamento o aggiornamento della tesi) solo se questa non è già stata assegnata ad un
    altro studente. Se risulta già assegnata a qualche studente l'operazione di assegnazione
    deve fallire. In caso di aggiornamento di assegnazione ad uno studente già presente nella
    mappa, la precedente tesi torna riassegnabile e deve essere inserita in lista */

    public ArrayList<String> laureandi () {...}
    //restituisce la lista ordinata delle matricole di studenti con tesi assegnata.

    public boolean laureato(String matr){...} // elimina dalla mappa lo studente (se
    presente) e la propria tesi.

    public ArrayList<Tesi> tesi_assegnate () {...}
    //restituisce la lista delle tesi già assegnate, ordinate in base al titolo.
}
```

ESERCIZIO 2 Realizzare un metodo ricorsivo interno alla classe **LinkedBinaryTree<>** che calcola l'altezza dell'albero binario corrente.

ESERCIZIO 3 Disegnare l'albero di ricerca 2-3-4 bilanciato risultante dall'inserimento della sequenza di chiavi (in questo ordine) **A L P I N B U S T E R Z O** in un albero inizialmente vuoto, usando il metodo di inserimento top-down. Trasformare l'albero risultante in un albero red-black.

ESERCIZIO 4 Realizzare un metodo interno alla classe **UnweightedNetwork<Vertex>** che restituisce la lista dei vertici del grafo in ordine crescente di distanza da un vertice sorgente dato in input.