Università Degli Studi dell'Aquila

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

Martedì 20 giugno 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.

Si considerino date le seguenti due classi (non sviluppare il codice), con i relativi metodi di accesso e un costruttore:

- Una classe **Studente** avente come variabili d'istanza il nome (stringa), il cognome (stringa), la matricola (intero univoco) e confrontabile secondo Comparable per cognome e in seconda istanza per nome.
- Una classe **Docente** avente come variabili d'istanza il nome (stringa) e il cognome (stringa)

Sviluppare le seguenti due classi:

1. Una classe **Tesi**, avente come variabili d'istanza un titolo (stringa) e un riferimento al docente proponente
2. La seguente classe **Assegnazioni** che gestisce le assegnazioni dei titoli di tesi ai laureandi:

public class Assegnazioni {

private ArrayList<Tesi> tesi; //lista di tesi proposte dai docenti (assegnate e non), senza duplicazioni di titoli

private TreeMap<Studente, Tesi> assegnazioni; // mappa degli studenti con tesi assegnata

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

public void **aggiungiAssegnazione**(String nome, String cognome, int matr, String titolo, Docente docente){...}

/* Se la tesi non è presente nella lista di tesi proposte, la si inserisce; se la tesi è già stata assegnata l'operazione fallisce, altrimenti si assegna/aggiorna la tesi assegnata allo studente specificato. In caso di aggiornamento di assegnazione, la precedente tesi resta in lista come libera e riassegnabile */

public ArrayList<Studente> **laureandi** (){...}

//restituisce la lista degli studenti con tesi assegnata, ordinata per cognome e in seconda istanza per nome.

public void **laureato**(int matr){...} // elimina dal sistema lo studente e la propria tesi

public int **disponibili**(){...}} // restituisce il numero di tesi presenti nel sistema non ancora assegnate

public ArrayList<Tesi> **tesiLibere** (){...} //restituisce la lista delle tesi non ancora assegnate

}

ESERCIZIO 2

Realizzare un metodo booleano interno alla classe **`LinkedBinaryTree<>`** che verifica se l'albero binario corrente è completo.

ESERCIZIO 3

Disegnare l'albero di ricerca 2-3-4 bilanciato risultante dall'inserimento della sequenza di chiavi (in questo ordine) **`T E R Z O A L P I N B U S`** 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 statico che, dato un grafo (di tipo **`Network<String>)`** specificato come parametro, restituisce una lista dei vertici del grafo in ordine crescente di distanza da un nodo sorgente. Il nodo sorgente è il vertice minore contenuto nel grafo.