Department of Computer Science

COMSATS Institute of Information Technology, Islamabad

Term Project for Data Structures

Section: BSE-3A & 3B   Session: Spring-18

Due Date: Monday April 16, 2018

Along, with this document you have been provided a .csv file containing a dataset of 5000 movies downloaded from IMDB. For each of the 5000 movies, details of 28 parameters have been provided like actors, directors, rating etc. You should use this dataset to create a database using various data structures studied in the class. To complete this project you need to define the below mentioned classes and also add the functionalities the details of which have been mentioned:

====================== MovieNode ===

A **MovieNode** Class which should

a) Store data of all the 28 fields for a movie.
b) Remember, some of the fields like actors, plot keywords etc. of the **MovieNode** class may contain more than one entries. Such entries should be stored in a relevant data structure like List/tree/hashtable. You should justify the choice of your data structure during viva exam.
   Note: Data of actors is already stored in the tree/list/hashtable constructed using objects of ActorNode class as data field. In case, two actors let's say X and Y have acted in the same movie, then, the references of the objects in which data of these actors is saved should be maintained in the MovieNode class in a list/tree/hashtable.
   Similarly, plot keywords are stored in a separate data structure. Only references to those plot keywords should be maintained in MovieNode

   ==

MovieAvlNode / MovieListNode class

You shall store data of 5000 movies in an appropriate linked list or in an **AVL tree**. For doing so, you need to define the following class

Class **MovieAvlNode**{

    MovieNode data;

    MovieAvlNode left;

    MovieAvlNode right;

}

Or

Class **MovieSinglyListNode**{

    MovieNode data;

    MovieSinglyListNode next;

}

Or

Class **MovieDoublyListNode**{

    MovieNode data;

    MovieDoublyListNode next;

    MovieDoublyListNode prev;

}

Note: The **key field** based on which nodes' logical position should be determined in the list or AVL tree is **movie's title** i.e. MovieAvlNodeObject.data.title.

This class should provide following functionalities:

a) A method which
    a. reads the .csv file line by line
    b. Creates an object of MovieSinglyListNode / MovieDoublyListNode / MovieAvlNode class for each movie
    c. automatically inserts all entries of that line in various fields of the MovieSinglyListNode / MovieDoublyListNode / MovieAvlNode class object and then
    d. inserts that movienode object in the list or avl tree

    Note: At the end of this method call,

    - an AVL tree consisting of 5000 nodes of type MovieSinglyListNode / MovieDoublyListNode / MovieAvlNode should have been constructed.
    - a Linked list consisting of 5000 nodes of type MovieSinglyListNode / MovieDoublyListNode should have been constructed.

b) Search movies by
    a. Genre
    b. Title
    c. Actors
    d. Rating range
    e. Year

f. director

======================================= Director Node ====================

DirectorNode class which should store

a) Director's name
b) The list of the movies he has directed.
   Hint: store references of the relevant objects of the MovieNode or MovieListNode
c) Provide functionality to
   a. Search directors by name. The search should return name of directors along with the details of the movies they have directed.

Note: You shall store data of all directors in a suitable linked list or AVL tree

=======================================Actor Node =====================

ActorNode class which should store

a) name of the actor
b) List of the movies in which he/she has acted. Remember, movies data is already stored in the linkedlist or avl tree constructed using nodes of type MovieNode. Thus, the list of actor's movies should contain only references to the MovieNodes in which he/she has acted.
   Note: You should decide whether the list of actor's movies should be stored in a linear linked list, in a BST, AVL tree or a HashTable.
c) The actor class should contain methods for the following functionalities
   i. Print the title of movies in which actor has acted along with some details of the movie.
   ii. Print co-actors in a particular movie
   iii. Print all co-actors
   iv. Any other relevant functionalities you feel should be provided.

=======================================

Guidelines (to be read very carefully!!):

1) One group shall have at max two members.
2) Evaluation shall be carried out via viva exam which may take 10 to 20 minutes per group.
    a. Each member of the group will be graded based on his contribution. It is possible that one group member gets 50/50 while the other gets 0/50!!
    b. Plagiarism cases will simply get zero marks.
3) In order to get maximum marks, you should use:
    a. Various data structures in this project like linked lists, trees, hash tables etc.
    b. Recursion for searching, insertion, deletion operations especially for traversing trees.
4) Use of array as data container is **prohibited** except in the cases where its requirement can be justified as in the case of Hash tables.
5) Use meaning names for variables, classes and methods.
6) All lines of codes should be properly commented.
7) Above each method's definition, a detailed description about its purpose and working principal should be provided.
8) The code must be properly indented.