TED UNIVERSITY

CMPE 223-01 Fall 2021

Programming Homework 3

Name Surname: Yakupguly Malikov

Section: 01

Problem Statement and Code Design

In this homework, I am going to create a database system in java. I am adding movie to the database which has movie title, release date, director name and actors with their name and roles in the movie. They all are stored in the Binary Search Tree (BST). Created Movie object for movies and Actor object for actors. Movie title release date and director name are stored in one BST by their movie release year and cast of each movie stored another BST by actor name. There are 5 classes which are Main, Actor, MovieObject, MovieDatabase and BinarySearchTreeCast. MovieDatabase is the BST of the Movie object and BinarySearchTreeCast is the BST of the casts which is Actor object

Implementation and Functionality

In the Main class, I added movie, printed all the movies, printed the details of given movie, removed the given movie, added actor to the given movie, printed the roles of the actors and printed the movies directed by the given director name.

In the MovieDatabase class, done the operations related to movie. In the addMovie() method adds the given movie to the BST. First, it searches for the movie node object in the BST whether the movie already exist in the BST. For this, I used findWithTitle() method and it iterates the BST and if it finds the movie with the given title in the BST it assigns the node object to nodeWithTitle. After, it checks whether the nodeWithTitle is null or not. If it is not null it means that there is already movie with given title. Then it deletes it and changes updated to true. Because I deleted the existing movie, now insert it in the correct order using insert() method. In the insert() method, it searches for the correct order by the release year of the movie in the BST recursively. After insertion, it prints that movie is updated if the value of updated is true or added to the BST if the value of updated is false. In the removeMovie() method, again it searches for the movie node object in the BST. If it cannot find the movie object, it prints that the movie does not exist. If it finds, it deletes using delete() method and prints that the movie has been deleted. In the addActor() method, it checks it the given movie exist in the BST. If not, it prints error message. If there is movie then it inserts it to the actor BST of the given movie. In the removeActor() method, like in the addActor() method it checks if the movie exists in the BST. If yes, it removes that actor from the actor BST of the given movie. In the showAllMovies() method it controls if the BST is null or not. If the BST is null, it prints none. If there is movie in the BST, then it prints the movies in ascending order according to their release year. In showMovie() method, like before it checks if the movie exists in the BST. If it exists in the BST, it prints the movie title, movie date, director of the movie and actors if there is any. In showActorRoles() method, it searches for the roles of the given actor name in any movie that exists in the BST. If there is at least 1 movie that actor played, then value of hasRole gets true in the findRoles() mehtod. Afterwards, it prints the roles the actor played in descending order according to the release year of the movie. In the showDirectorMovies() method, it searches for the movies that is directed by the given director. If there is at least 1 movie which is directed with the given director, then the value of hasMovie changed to true. The movies directed by him/her is printed according to the release year of the movie

in descending order. In the minValue() method, it finds the minimum movie release year from the given subBST or BST.

In the BinarySearchTreeCast class, done the operations related to actor. In the insertActor() method, adds the actor to the BST. First, it checks whether the actor exists in the BST. I used search() method and it goes through each node in BST. If it finds the actor with the given name, return its actor node object. If the returned node object is not null, it means that there is already actor with the given name then deletes it and changes the value of updated to true. Because I deleted the already existing actor, now I insert it with the new role. After, if it is updated then it prints that the actor role of the given actor is updated. Otherwise, it prints that the given actor has been added to the given movie. In the insert() method, it inserts the given actor to the actor BST. It inserts the actor which fits well according to his/her name. In finding() method, it finds the actors with the given name. in the findRoles() method, it prints the role and the movie of the given actor played and changes the value of the hasRole to true. In setRoot(), assigns the given root to the root. On the other hand. getRoot() method returns the root. inOrder() method, prints all actors in the BST in order to ascending order. Lastly, minValue() method, finds the minimum actor name from the given subBST or BST.

Actor class creates the actor object with the name and role of the actor. First, assigns the parameter values to the relevant values of the object. getName() method returns the full name of the actor. getActorRole() method returns the role of the actor in the movie. setName() method assigns the full name of the actor to the name.

Movie class creates movie object with the movie title, director name and the release date. First, assigns the value of parameter values to the relevant values of the object. After, it sets the release date of movie with the given release day, month and year. getDate() method returns the release date of the movie. setDate() method assigns the release date of the movie to the date. getReleaseYear(), getReleaseMonth() and getReleaseDay() methods respectively returns the release year, release month and the release day of the movie. getDirector() method returns the full name of the director of the movie. getMovieTitle() returns the title of the movie.

Testing

My testing takes place in the Main class. In main class, I added movie to the BST, printed all the movies from the BST, removed specified movie, printed the details of the specified movie, added actor to the specified movie, removed actor from the specified movie, printed all roles played by the specified actor in any movie in the BST and printed movies directed by the specified director.

Final assessment

I liked the homework, and it was my first experience solving problems like this. Homework itself was challenging. It is not hard to solve but because there are more code lines, it got complex to follow. By solving this problem, I learned the BST structure, how to add, remove or update the in BST. Most importantly, I learned the nested BST which BST inside BST.