Jonathon Harris

CS202

11/28/20

Media Manager – Efficiency Writeup

The main purpose of project 5 was to enhance upon project 4’s design by creating a binary search tree of courses. So, the overall data structure is a binary search tree where each node in the tree has an array of linear linked lists. Course has a course name and course ID data members. Instead of creating a node I made course act as its own node so it has a left and right reference pointing to another course and my tree has a course root reference. For my tree functionality I had an add course, remove course, remove all courses, display all media, and a find course method.

For my menu options I had add options which included add course or add media to course. My remove options were to remove a course or remove media from a course. My display options included: display all courses, display all media, display individual media and display all by topic. Finally, I had an import media from text option which prompted the user for a specific course, then they could choose which type of media they want to add.

A strange phenomenon I found with java is when you need to remove all the nodes of a tree you can just set this root to null. This felt wrong and is very different from C++. Also, when adding a node to a tree I noticed you need to have a special case for if this root is null then add to this root, otherwise If you just add to root then it will not work. The hardest part of project 5 was the subtle intricacies of java. But overall, this assignment was relatively easy. Since most of the work was done in project 4 a big part of project 5 was just making wrapper functions for my tree’s courses to use the playlist functions, once that was finished it was pretty much done. As far as my efficiency for this assignment for my removal function had the worst efficiency with O(n) time complexity. My search and add function both have O(h) time complexity where h represents the height of the tree. Display is also O(n) time complexity. There was a design option that I went with, I decided to make the menu part of main as opposed to creating a class and function for it.

I really enjoyed intellij for this project it was really intuitive and easy to use. One of the best features is the instant ability to see errors and potential errors. When using the debugging software I noticed you could click on a variable or object and it would open a pane window up to show you all the data members and what they are set to. This was extremely helpful when debugging for my add function which took awhile to get working. Initially I had my add function return an integer but I couldn’t get the function to work. So I changed it to return a course object and then for the function call I had to set next with the function call get next. Once I did that then the add course function started working. Another issue I had was with a display all function, the function would only display one. When I stepped through it with the debugger I found that root was pointing to another object but it would just return without going to the next node. After an hour or two I realized that my function call wasn’t happening instead it was just moving to root’s left and then returning so debugging showed me that the add function was working because there was something that root was pointing to which meant it was the display function.