Project #1

Binary Search Tree

CS 241 Section 1

Matt Dohlen

[matt.dohlen@gmail.com](mailto:matt.dohlen@gmail.com)

4/22/2014

Section 1. Project description

The binary search tree program should read from the standard input a sequence of integer values, with each value separated by a space. It should build a binary search tree using these values in the order they are entered, print 3 traversals: pre-, in-, and post-order, allow the user to insert/delete a value and also allow the user to find the predecessor and successor of a value.

Section 2. Project specification

The binary search tree program require a main method and two classes. One class is the class is the search tree itself and the other is for each node in the search tree. The tree class has methods for inserting, deleting, and finding the successor and predecessor. The node class has various method for altering, finding, traversing, and determining aspects of the node. These classes allow the user to enter data into a search tree which properly orders the data.

Section 3. Testing methodology

This project required vigorous testing. The method I used was to basically test users input from the first enterance to the last. First I would enter the string of integers that would initialize the tree. This required me to test the BST's insert function for any fault, which did happen and did need to be fixed. Because I have never coded in Java before I was passing a lot of parameters by reference which is not possible like in my native language of C++. Realizing this mistake I knew I would have to go back and fix many other functions that I was doing this in. The next step in my testing was to go through each option in the menu of options for the user and test each option. With the insert option already tested it was time for deletion. I would enter values that were and were not in the tree to see various outputs. For successor and predecessor I used similar methods but I also attempted to check the highest and lowest values in the tree. After checking each individual function I would test all the functions simultaneously in various orders to see if outputs varied much.

Section 4. Lessons learned

Since this is basically my first time coding higher level project in Java I learned a considerable amount. I have taken a Data structures class in C++ at the college I transferred from but have only taken CS 140 to learn Java. I did not learn much about coding a BST but I learned a ton about the syntax of Java. For one I learned that it is not truly possible to pass a value by reference like in C++. I learned but was not entirely able to use generic values (I switched to integers half way through). The major thing I learned was how to create classes in Java which is very different from using a header file in C++. The most difficult thing to work around was passing by reference because I'm so used to using that and pointers. All in all it was a great learning oppurtunity for me.

Section 5. Analysis of output

When I put in the input that was given to us it comes out perfectly but there are a few problems when I enter things in other ways. Occasionaly the output for the predeccessor shows data that has already been deleted out of the tree. This is the only major problem I found with inputting in to the tree but there is also other problems with entering data. I didn't use very many measures for parsing the users input which means the data needs to be entered perfectly or the program fails.