Katrina Slivkoff

Professor Potika

CS 146

9 December 2019

Project 4

I. Introduction

This project enhanced my understanding of red-black-trees. At first, this was a difficult concept for me to grasp. I could not figure out whether a node would be red or black or where to add any of the other nodes. Luckily, Professor Potika provided us with a starter code and a tester code so we could check our work along the way.

II. RedBlackTree Class

The first method that we had to implement ourselves was the “addNode” method. This method placed a new node in the tree and colored it red. First, you checked if the node was empty. If it was not empty, then you could update the node. After the node was updated, the “fixTree” method was called.

The “fixTree” method rotated the node where it was supposed to be and made it the right color of either red or black. It called the methods “rotateRight” and “rotateLeft” so that the nodes could rotate the correct way. It also called the methods to determine if the node was a left child, right child, aunt, or grandparent node.

The “rotateRight” and “rotateLeft” methods determined where to put the node in to the tree by rotating it to the right or to the left. These are similar methods and followed the rotate methods we learned in class.

III. Conclusion

Overall, this project was relatively easy for me to understand. The hardest part were the JUNIT tests. I could not figure out how to do the spell checker within the JUNIT test, so I ended up making my own test that printed out which words were spelled wrong. However, since Professor Potika provided one test, this JUNIT test passed my program.