Omid Sharghi

CS 146 Section 1

Programming Project 3

After running the tests, the measured time to create a dictionary of 699,800 words ranged between 143 milliseconds to 153 milliseconds. Looking up values in the dictionary was at most 1 millisecond, but the majority of the time it took less than 1 millisecond. For the dictionary, I created a separate class called RBFileLoader, which looks for a .txt file in the parent directory. If a .txt file is found, it reads each line using the BufferedReader and inserts each line into a Red Black Tree object, all while being timed.

To make the testing process more straightforward, one of the changes I made to the original RedBlackTree.java file was removing the static keyword from the root instance variable. When the root variable was static, the Red Black Tree node data would persist every time I created a new instance of the Red Black Tree. With the static keyword, testing tended to be problematic since every method in the RedBlackTree class was centered around the root.

To avoid null pointers, I created an instance variable titled nilNode, which was a Node object, but the data was an empty string. I set the parent of the root equal to the nilNode and all leaves pointed to the nilNode.