Kevin Rodriguez

LAB 4 Report

***Introduction***

For this lab, I was given a file with over 350,000 English words. The main purpose of this lab was to create a function that will read all of these words and store each word in a binary search tree. We had to implement two types of trees, which were AVL Trees or Red-Black Trees. We need to prompt the user to a menu to decide what kind of tree they want to use. The appropriate tree is then created based on their answer. I had to create another function that will return the number of permutations a word has. Each permutation should be in the tree created. The final part of this lab was to create a function that would receive a file and return the word in the file that has the highest number of permutations.

***Solution/Design Implementation***

(1) The way I designed the solution for this lab was to create a menu that will be prompted to the user and let them decide what kind of tree they want (AVL or RBT). I created a different method for each of the trees that read the file and insert each word in the tree. The reasoning behind this is because each tree has a different type of insert method due to the properties of each tree. For example, in the Red Black Tree, you need to keep track of the color of each node and in the AVL Tree there is no color involved. Since both of these type of trees are a binary search tree, all three main methods (insert, search, delete) are in O(log n).

(2) For the function that returns the number of permutations a word has, I created a global variable called ‘count’ that updates every time a permutated word was found in the tree.

(3) For the last function created, I simply created 3 global variables. One called “counter” which keeps track of how many permutated words are made and are in the file of a specific word. Next global variables are called “max\_count & max\_word which keeps track of the maximum number of permutated words and the word that has the maximum number of permutated words.

***Experimental Results***

In order to test my program, I created a dummy file with some random English words with no permutations except with the word spot. I added 5 permutations from the word ‘spot’ in the file in order to see if all my methods work. In fact, whenever I run my program, the word with most permutation is spot.

***Conclusion***

In this lab, the main thing that I learned is how an AVL Tree and an RBT is implemented in coding. The concept was learned in class, but the implementation was used more in depth in this lab. I also learned a new thing in python and that is how to use global variables to your advantage as well as importing different files into my main file, so I could organize my code into having each class of the tree to be in one file.

***Signed Academic Honesty Certification***

I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class