DATA STRUCTURES PROJECT 2

CSE2225

Name=Umut

Surname= Bayar

Number=150120043

Due Date= December 28, 2022

Problem= In this project, you are required to write a program to compare the performance of AVL and Splay trees based on the two criteria: the total number of comparisons and the number of rotations.

Before I started writing code, I thought how we could solve the problem. I need to create a sketch for the Project. First of all I need to read the data. Then I have to correctly put the data on the tree by comparing each data and i read with the previous ones. As long as the balance value fort he AVL tree is greater than 1 or less than -1, I can continue to insert without a problem. If it does not comply with these values, I can stabilize the tree with the necessary rotation functions. For the splay tree I need to process every added value.I will send each inserted value to a function. This will bring the newly inserted value to the root after necessary rotation operations. Finally with the counters I will put, i will find the number of rotations and comparisons for both the AVL tree and the Splay tree separately.

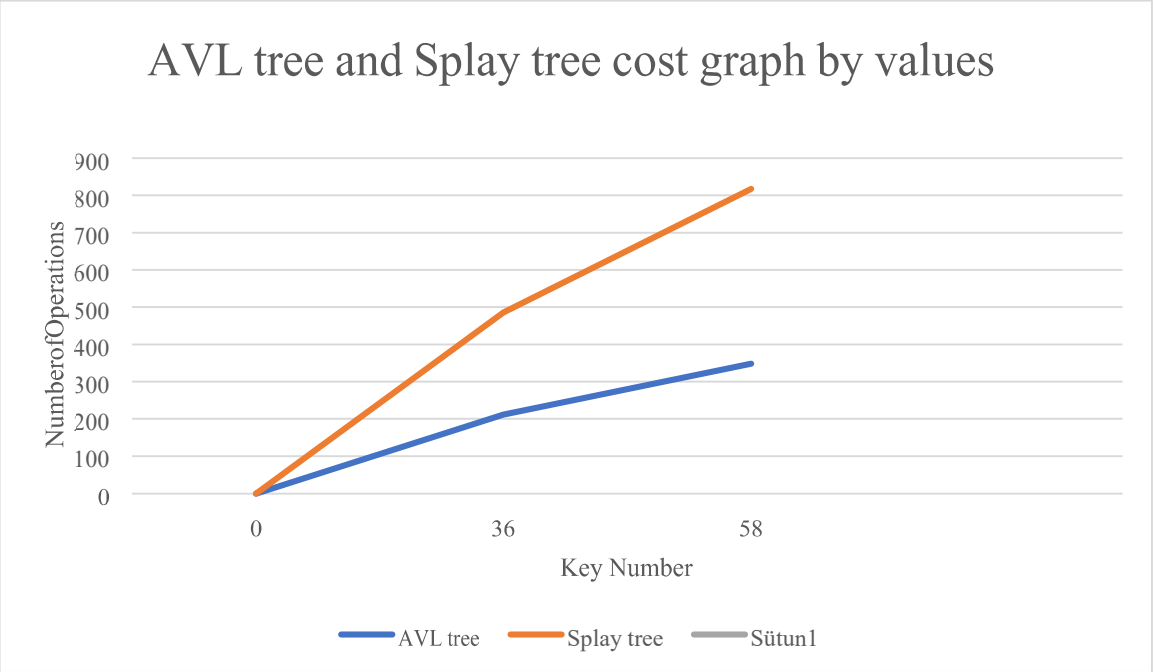
First of all, I created a struct named Node, a key variable that stores the current data, and a height variable that holds the height of the tree. Then I reserved a place in memory. I set the left, right and current value of the node to null. Now it's time to read the values. For this, I first opened the file inside the main function. Then I checked if the file is empty and if not, I sent the values in the text file first to the AVL function and then to the Splay function.I

also printed the necessary prints inside the main function to finally print the total cost i need to calculate. I have defined 4 new variables at the top for these. These are rotateNumberAvl, rotateNumberSplay,compareNumberAvl and compareNumberAvl. For now, we just created our variables without any action. Now i need to write a function for AVL tree. This will keep the value read from the file in the key variable and place it in the nodes. Before determining where to place the newly imported value, the same operations must be done for the splay tree. For this, we create a new function called splay with the same variables as the AVL tree.

Now I have created left Rotate function and right Rotate function for rotation operations, height function to find the length of the tree, and finally bigOne function to find which value is bigger. I could write functions for left left ,right right ,right left and left right separately, but instead I got these states by sequentially calling leftRotate and rightRotate. also these worked for zigzag, zigzag zagzig and zagzag situations. Finally, I created an preorder function to call the tree preorder(N,L,R).

Now we have completed the functions. We must calculate the total cost by using the rotate Number Avl, rotate NumberSplay, compareNumberAvl and compareNumberSplay variables that we defined earlier. While the cost of single rotations and two-node comparison is 1 tus, the cost of double rotation should be calculated as 2 tus. Since it will call the rotation function every time the condition is met, I have calculated the total cost for both the AVL tree and the splay tree by increasing the value of the variable each time the condition is met. Finally, we printed the tree with the values we calculated and

preorder(N,L,R) sequentially. On the last page, there is the value/cost graph that will enable us to understand which tree we should prefer in which situation.



For 36 keys, the AVL tree has 212 operations, while the splay tree has 486 operation numbers. For the key number 58, the AVL tree has the operation number of 349, while the Splay tree has the operation number of 818.

As can be seen in the graph, the Splay tree becomes more costly as the number of keys increases. So its performance is lower