CSE2225 DATA STRUCTURES SECOND PROJECT REPORT

Student Name: Barış Giray Akman

Student ID: 150121822

Content Table

Overview

In that project, I have written splay tree and AVL Tree code. After implementation of both, I have compared their performances accordingly with their costs. In the project, cost means sum of count of total rotation and count of total comparison during the insertion.

My Approach

My approach for AVL tree is recursively going downwards through the tree and finding the optimal location for our new node. Until here, things are not tricky. By then, I calculated height values for each node recursively. If there is an imbalance for any node, then I have rotated the nodes according to that rule. I have done that until going up to the root node along the parent nodes.

Instance variables I have used for AVL Tree is, height, val variable for representing the numerical value of a node, and lastly left and right child of a node.

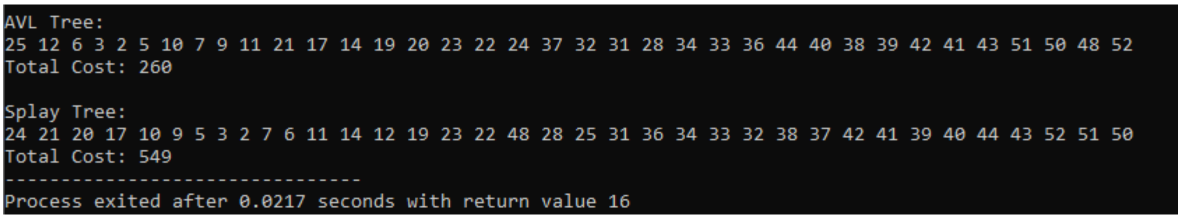
For Splay Tree, I have followed a different strategy. Initially, I have inserted the new node to splay tree if node doesn't exist in the splay tree. If it exists, I didn't insert it.

Then, I have splayed the new node until bringing the new node as a root of whole tree. Even if I didn't insert a new node(That inserting a node that is already exist in the tree) I have brought it as a root of whole tree.

Outputs and Comparison

Outputs that I have achieved with sample outputs are as follows in the screenshots:

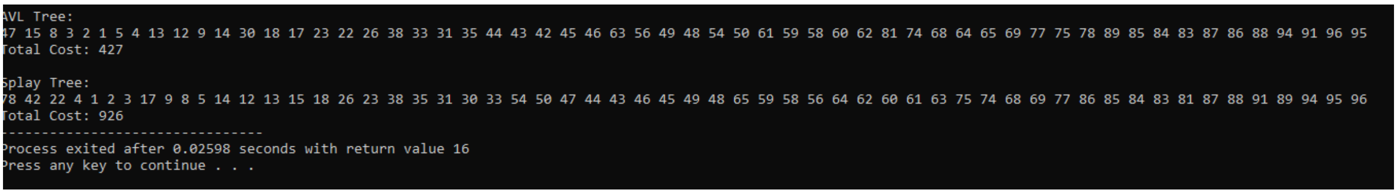
For the first input file(input1.txt):



```
AVL Tree:
25 12 6 3 2 5 10 7 9 11 21 17 14 19 20 23 22 24 37 32 31 28 34 33 36 44 40 38 39 42 41 43 51 50 48 52
Total Cost: 260

Splay Tree:
24 21 20 17 10 9 5 3 2 7 6 11 14 12 19 23 22 48 28 25 31 36 34 33 32 38 37 42 41 39 40 44 43 52 51 50
Total Cost: 549
--------------------------------
Process exited after 0.0217 seconds with return value 16
```

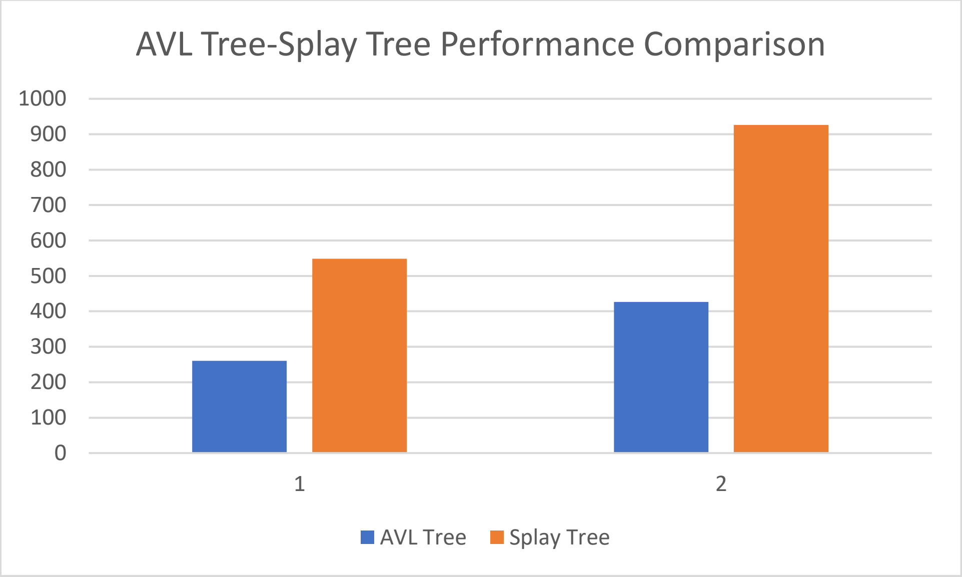
For the second input file(input2.txt):



```
AVL Tree:
47 15 8 3 2 1 5 4 13 12 9 14 30 18 17 23 22 26 38 33 31 35 44 43 42 45 46 63 56 49 48 54 50 61 59 58 60 62 81 74 68 64 65 69 77 75 78 89 85 84 83 87 86 88 94 91 96 95
Total Cost: 427

Splay Tree:
78 42 22 4 1 2 3 17 9 8 5 14 12 13 15 18 26 23 38 35 31 30 33 54 50 47 44 43 46 45 49 48 65 59 58 56 64 62 60 61 63 75 74 68 69 77 86 85 84 83 81 87 88 91 89 94 95 96
Total Cost: 926
--------------------------------
Process exited after 0.02598 seconds with return value 16
Press any key to continue . . .
```

Moreover, I have graphed the results. It can be checked in the next page:



As it is seen for both input files, AVL Tree performs better than Splay Tree because of lesser cost value of AVL Tree.

Result

As a result, I have implemented AVL Tree and Splay Tree. I have seen whole cases for each situation while coding those.