<u>COMPARISON OF PARAMETRE BETWEEN</u>

<u>AVL ,BST AND TREAP</u>

Roll number :-214101022

- Five text files are generated using test case generator program.
- In each text file 10,000 key values are present which is randomly generated.
- In every text file the range of numbers is different.
- Now we will run AVL, BST and TREAP using these text files and compare parameters.

Output of AVL code:

```
Insert: 70%     Delete: 30%     file1.txt
----------------------------------------------------------------------------------------------------
File            Tree_height     Rotations       key_comparisons         Avg_node_height
----------------------------------------------------------------------------------------------------
AVL:            11              2253            94376                   2.32258

Insert: 75%     Delete: 25%     file2.txt
----------------------------------------------------------------------------------------------------
File            Tree_height     Rotations       key_comparisons         Avg_node_height
----------------------------------------------------------------------------------------------------
AVL:            13              2733            107487                  2.31755

Insert: 65%     Delete: 35%     file3.txt
----------------------------------------------------------------------------------------------------
File            Tree_height     Rotations       key_comparisons         Avg_node_height
----------------------------------------------------------------------------------------------------
AVL:            12              2577            102046                  2.3353

Insert: 60%     Delete: 40%     file4.txt
----------------------------------------------------------------------------------------------------
File            Tree_height     Rotations       key_comparisons         Avg_node_height
----------------------------------------------------------------------------------------------------
AVL:            13              3198            110589                  2.32092

Insert: 80%     Delete: 20%     file5.txt
----------------------------------------------------------------------------------------------------
File            Tree_height     Rotations       key_comparisons         Avg_node_height
----------------------------------------------------------------------------------------------------
AVL:            14              3073            110876                  2.32516
```

Insertion and deletion percentage is different for every file.

We can see final tree height, number of rotations, key comparisons and avg. node height for each text file.

Now we are going to compare these parameters with BST and TREAP.

Output of BST code:

Insert: 70%   Delete: 30%   file1.txt

File	Tree_height	Rotations	key_comparisons	Avg_node_height
BST:	19	0	101719	3.27165

Insert: 75%   Delete: 25%   file2.txt

File	Tree_height	Rotations	key_comparisons	Avg_node_height
BST:	23	0	135163	3.29938

Insert: 65%   Delete: 35%   file3.txt

File	Tree_height	Rotations	key_comparisons	Avg_node_height
BST:	22	0	115139	3.19003

Insert: 60%   Delete: 40%   file4.txt

File	Tree_height	Rotations	key_comparisons	Avg_node_height
BST:	23	0	127633	3.32805

Insert: 80%   Delete: 20%   file5.txt

File	Tree_height	Rotations	key_comparisons	Avg_node_height
BST:	27	0	129955	3.39769

Output of TREAP code:

Insert: 70%   Delete: 30%   file1.txt

DS	Tree_height	Rotations	key_comparisons	Avg_node_height
Treap:	23	6518	104325	3.56706

Insert: 75%   Delete: 25%   file2.txt

DS	Tree_height	Rotations	key_comparisons	Avg_node_height
Treap:	23	8083	126806	3.45863

Insert: 65%   Delete: 35%   file3.txt

DS	Tree_height	Rotations	key_comparisons	Avg_node_height
Treap:	21	7970	118065	3.46706

Insert: 60%   Delete: 40%   file4.txt

DS	Tree_height	Rotations	key_comparisons	Avg_node_height
Treap:	33	9244	132633	3.5733

Insert: 80%   Delete: 20%   file5.txt

DS	Tree_height	Rotations	key_comparisons	Avg_node_height
Treap:	27	8661	129392	3.43728

(1) HEIGHT OF THE FINAL TREE :

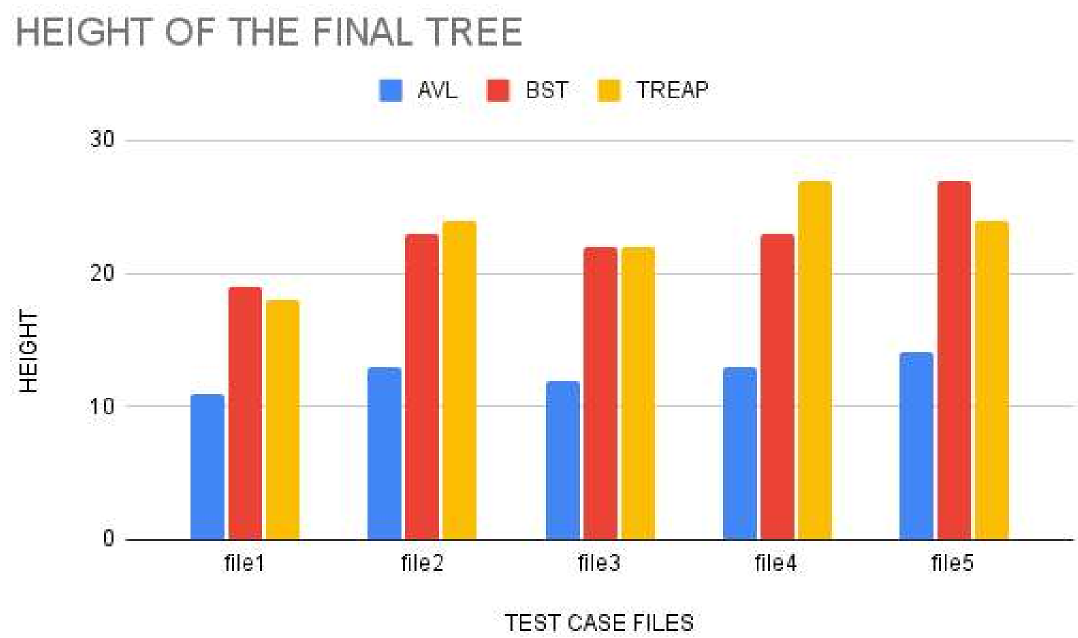

Chart of final tree height comparison

- Height of AVL tree is log(n). We can see the heights are in range 11-14 for 10,000 key values. Which makes it same as theoretical.
- BST and TREAP height will be in between log(n) and n.(n= 10,000)
- AVL gives best performance here. Here we can that height of AVL tree is less than BST and TREAP.
- Height of TREAP depends on its priority values, which is random. So we cannot compare the height of TREAP with BST. It will be more sometimes and less sometimes.
- Final height of AVL and BST in constant for these 5 test case files.
- Final height of TREAP is not constant for these 5 test case files because of the priority values.
- Priority values are randomly generated that’s why height will change each time you will run the program.

(2) <u>NUMBER OF ROTATIONS OF THE FINAL TREE</u> :

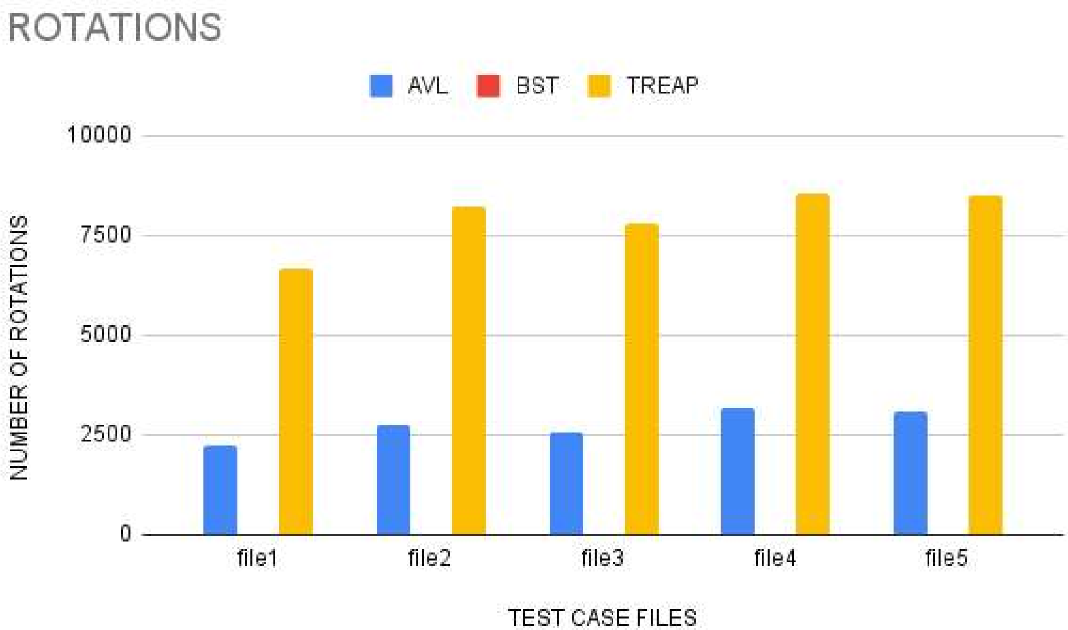

Chart of rotations(insertion + deletion) comparison

- There are no rotations in BST. Therefore values are zero in all five files.
- As the height of the AVL tree is always less than or equal to TREAP, the rotations takes place in AVL will also be less than or equal to TREAP.
- Rotations depends on input key value for AVL.
- Rotations depends on priority value for TREAP.
- As priority values are randomly generated the number of rotations are not fix for given test case files.
- Every time we run the program we will get new number of rotations.

(3) NUMBER OF KEY COMPARISONS OF THE FINAL TREE :

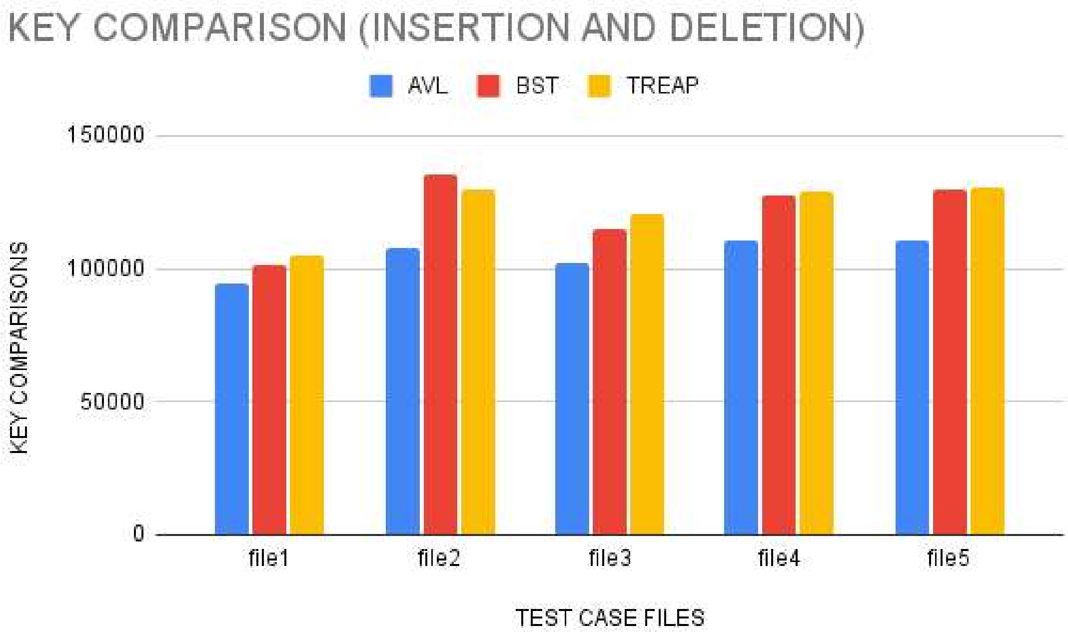

Chart of key(insertion + deletion) comparison

- Key comparisons totally depends on height of the tree.
- Key comparison = n*log(n) = 10,000 * log(10,000)
- Key comparison values are not exactly the same but the difference is not very big.
- AVL tree is always balanced. So the number of comparisons will be little lesser than other Data structures.
- As height of the TREAP totally depends on priority values, we cannot compare it with BST.
- BST height totally depends on input key values. Therefore we cannot compare it with TREAP.

(4) AVERAGE NODE HEIGHT OF THE FINAL TREE :

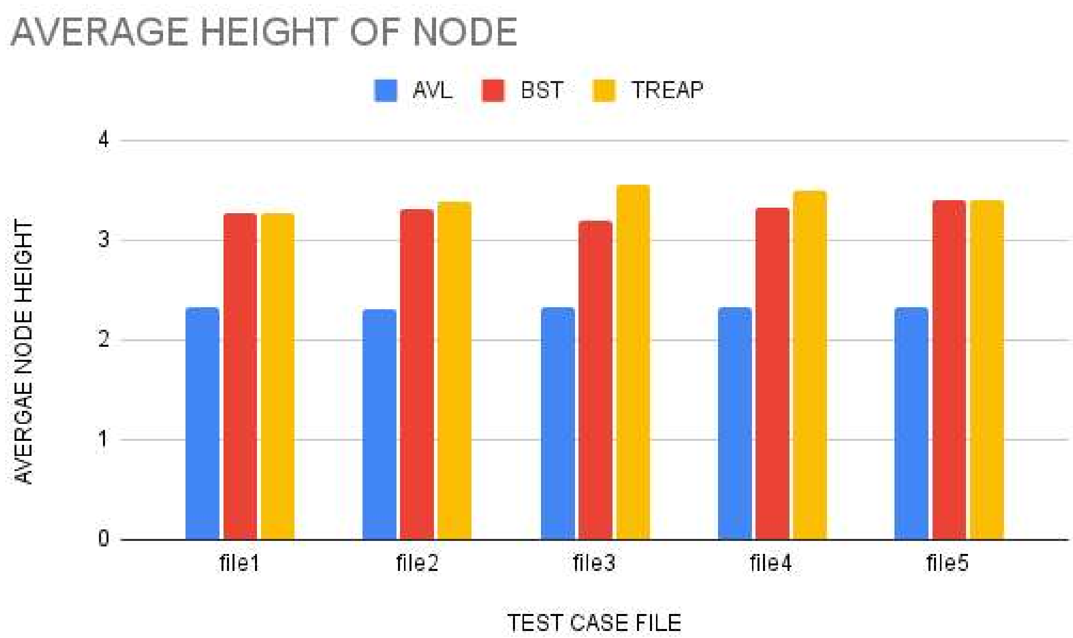

Chart of average node height comparison

- AVL tree is always balanced. So the height will always be little lesser than other Data structures.
- Height of leaf node = 1.
- Height of root = longest path distance from root to node.
- Therefore AVL average node height is always less.
- BST average node height is less here than TREAP. It is not necessarily true for all other test files.
- In some cases average node height of TREAP can be less than BST.