**Assignment 4**

|  |  |  |
| --- | --- | --- |
| ~Ankur Kumar |  | 210152 |

**Explanation:**

Here we are adding numbers in BST in increasing order.

So my code firstly does a right traversal and adds the node(in the process increasing the size of all the nodes in the path)

Then we go through the same path again via right traversal to check if any node has imbalance **(size subtree>3/4 size parent)**

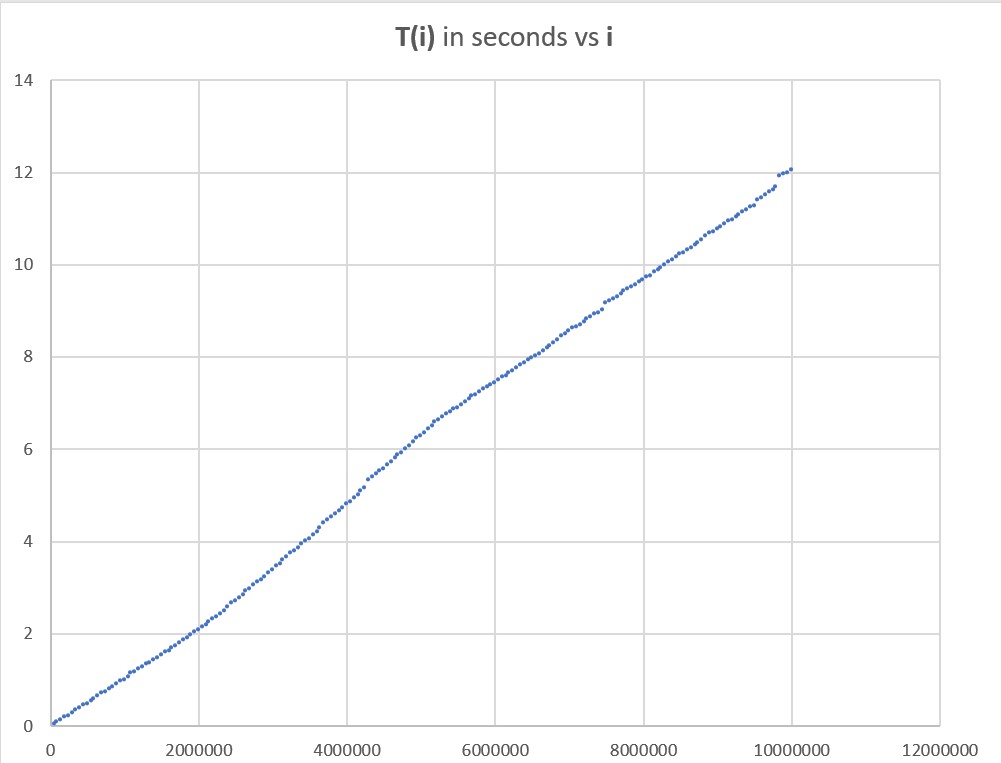
If it has any imbalance we use the balance() function to create a sorted array of the BST node pointers and then with help of recursion create a perfectly balanced BST (we just changed the pointers and saved time required to free nodes)

Also after an insertion there may have been imbalances at multiple nodes…therefore we rearranged from the root of tree to make sure all of them are dealt with

(We had another idea and we worked on it too… that all imbalances(if any) must occur in the path of insertion…so all such nodes must be present in that branch only…so we right traverse from root and look at first such node where imbalance occurs) after we get that we rearrange everything below that node(since everything above and left is balanced and this takes care of imbalances further down the path (if any)

But this code was stuck in some segmentation fault error we couldn’t debug…so we went foreward with root balance approach above).

**Graph 1: Time taken vs nlogn**



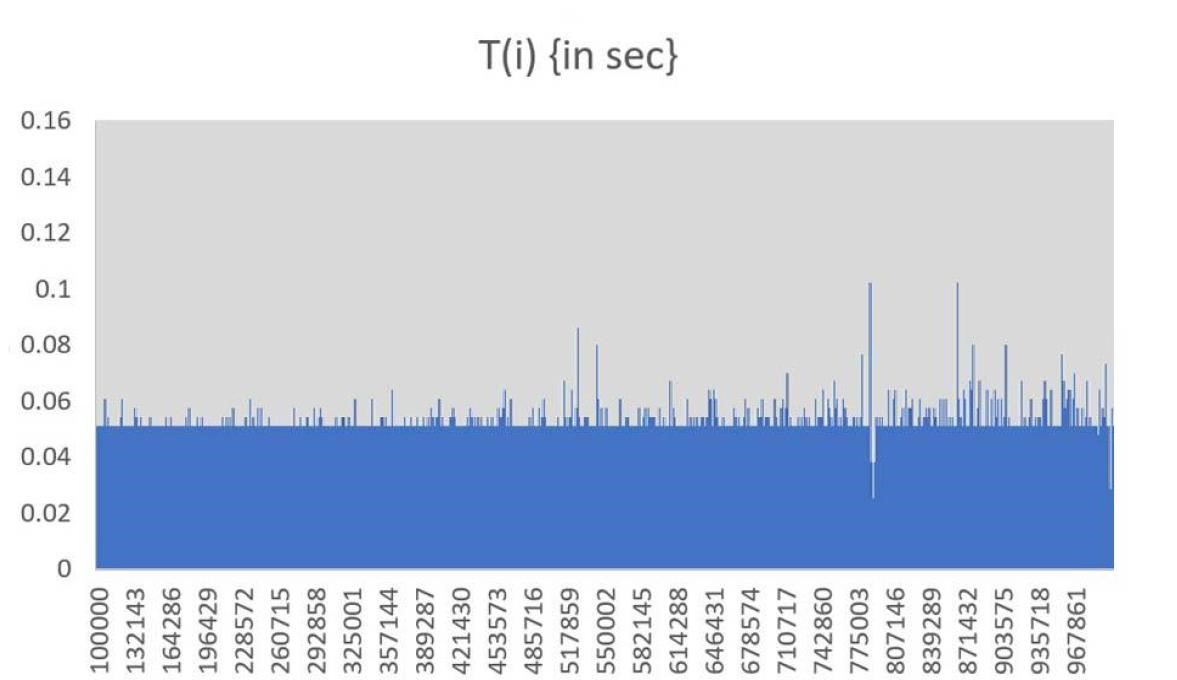
Here we can clearly see that the graph is linear in nature denoting that out algorithm has time complexity of order O(nlogn).

We also see frequent jumps(discontinuity) in the graph

These can be explained by the fact that the algo sometimes rearranges the tree midway insertion and the time taken is taken in account while calculating T(i).

So the jumps correspond to the insertions where imbalance occurs and the tree had to rearrange/balance itself which took more time than the usual case.

**Graph 2: Time per iteration**



This graph further supports the point that almost all iterations take same time and hence linear nature of graph…except some of the iterations which take quite longer than expected which is accounted by the fact that Balancing of tree occurred in that particular iteration.