Analysis Explanation.

BST

For Large and Small File

Sorted.txt

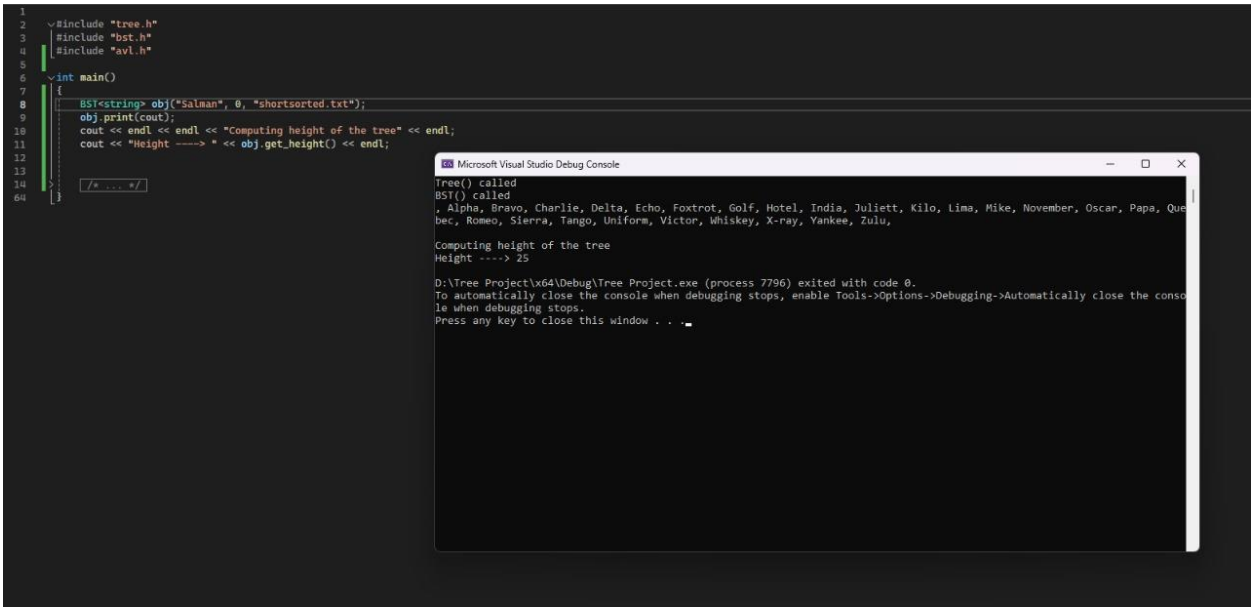
When I try to run the code on sorted.txt using a BST, it goes two ways. Either it throws an exception on stack overflow or it generates a tree which is either left skewed or right skewed with a height of 45000+.

This is the worst case of this data structure as it is now behaving as a Linked List instead of a Tree.

Its searching operation will now take O(N) instead of Log(N).

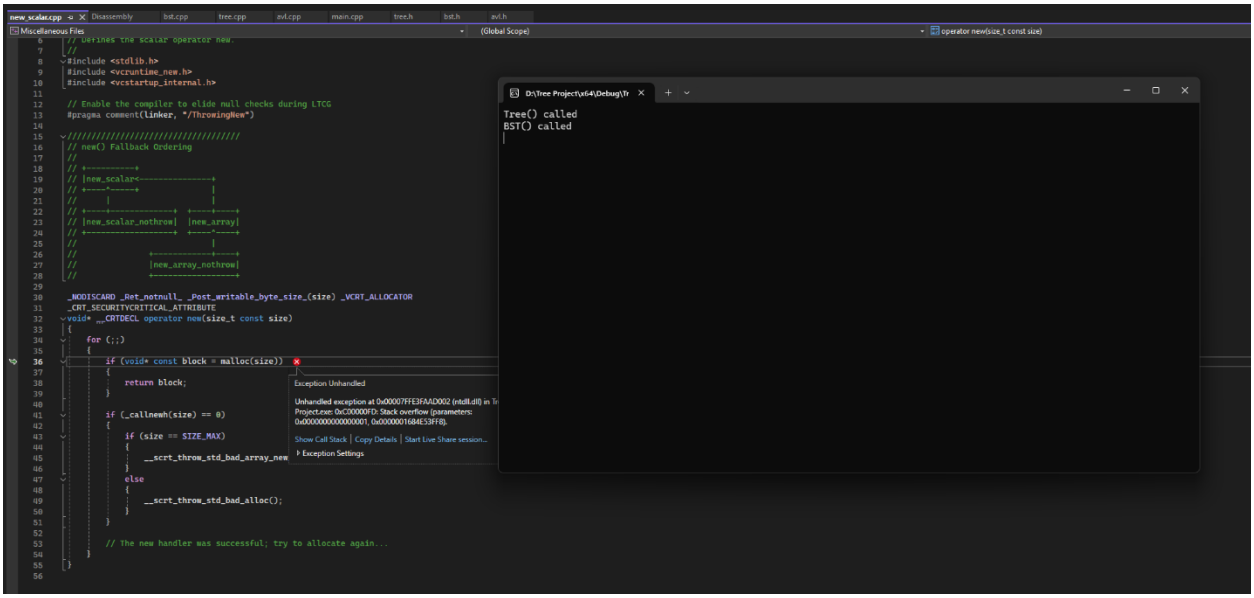
Examples:

Shortsorted.txt



As you can clearly see that height of our tree is, 25 for a total count of 25 Nodes. It indicates that tree is skewed to only one side and will take O(N) instead of Log(n).

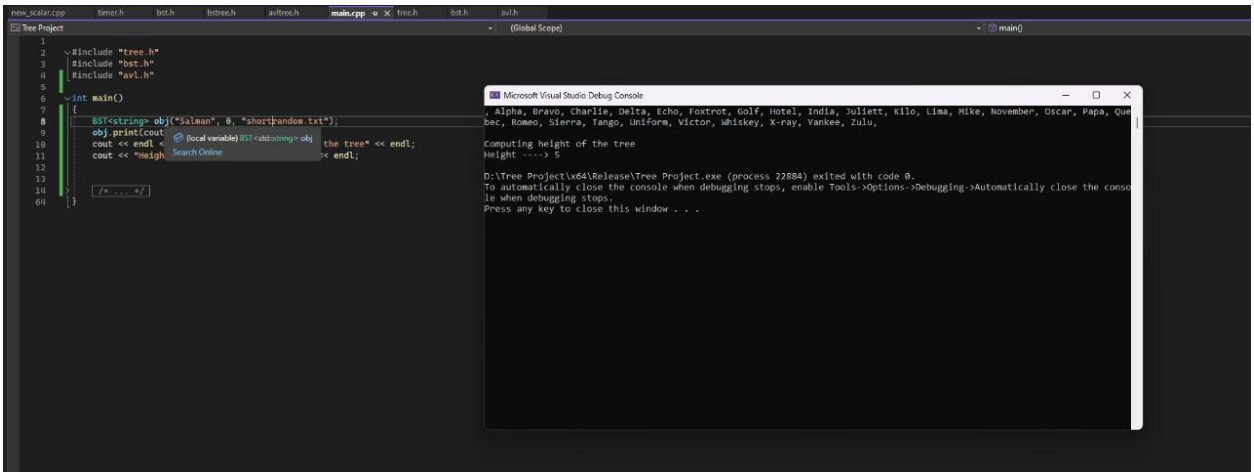
Sorted.txt



For this dataset, compiler throws an StackOverflow error, because tree is just growing in one direction and behaving like a linked list. For this case our height would be 45393 and Count will also be 45393.

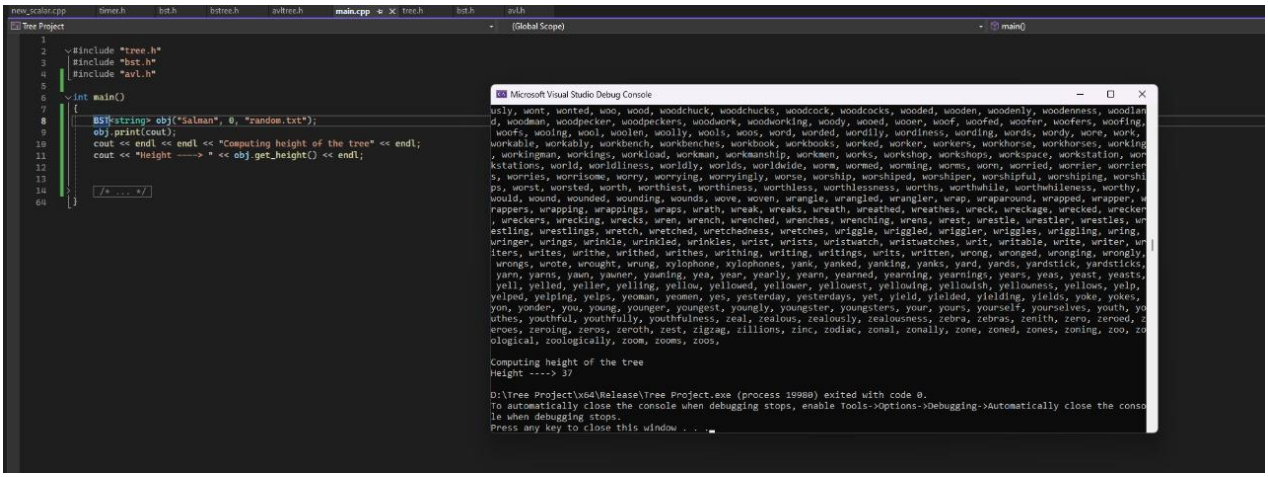
This indicates the **limitations of a BST Tree when data is sorted.**

ShortRandom.txt



For this purpose, BST has a height of 5 with a total node count of 25. As data is random, tree grows on both sides and achieve a height of 5 on which search operation would take place as Log(n).

Random.txt



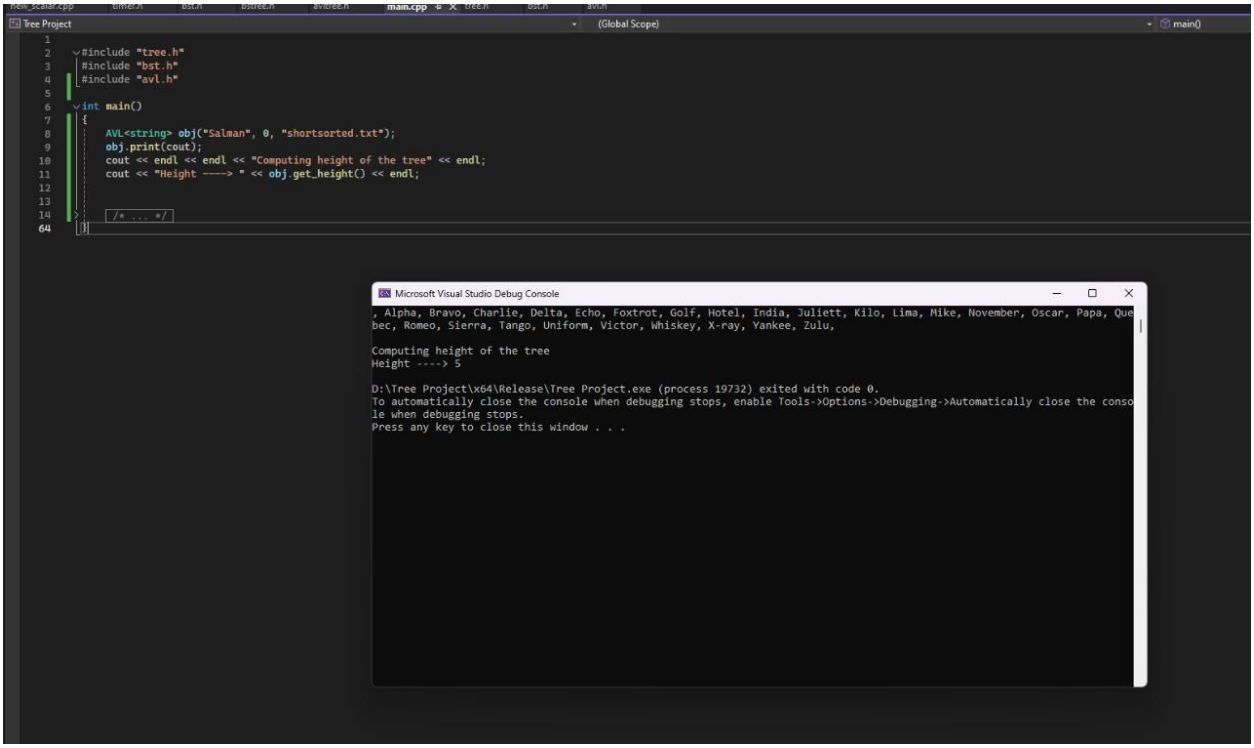
For this purpose, as data is random BST has a height of 37 with a total node count of 45393. This shows the power of BST when data is random. For this node count if we want to search anything, it would happen in Log(n).

AVL

For Large and Small Files.

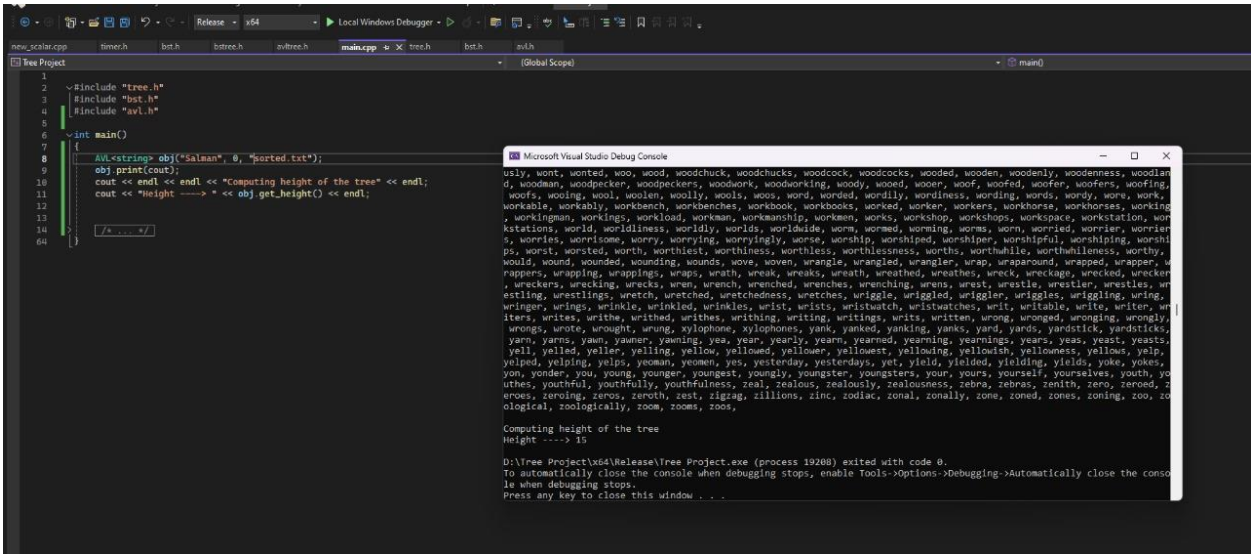
Examples.

ShortSorted.txt



As you can clearly see that height of our tree is, 5 for a total count of 25 Nodes. It indicates that tree is self-balancing itself and will perform Log(N) on search operations.

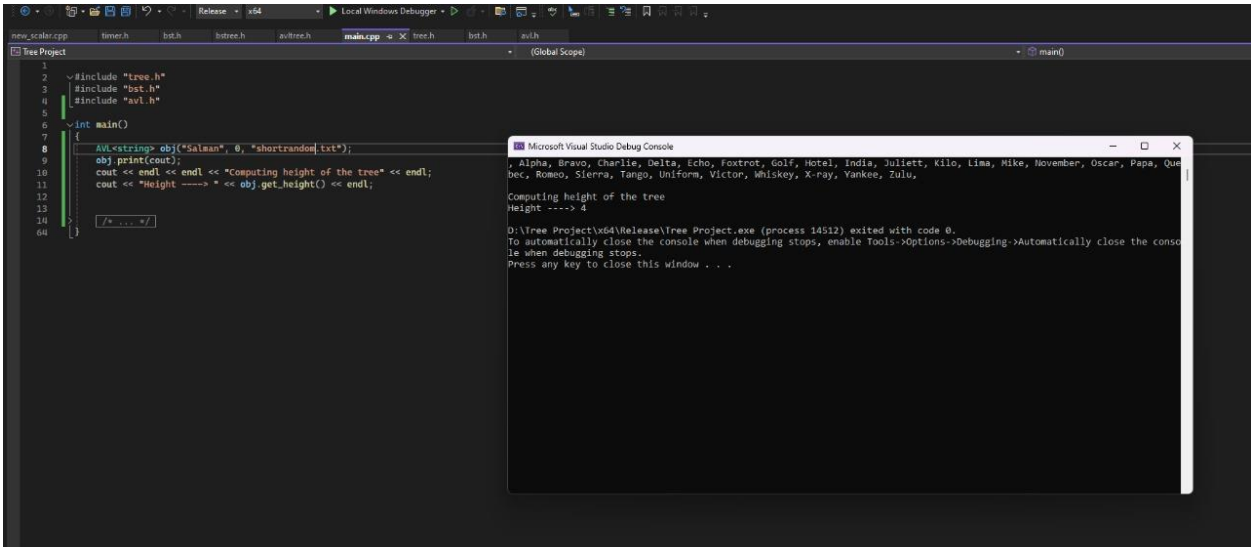
Sorted.txt



For this dataset, AVL expands to a height of 15 for a total of 45393 nodes. This result is remarkably good and will give us any Node in Log(N).

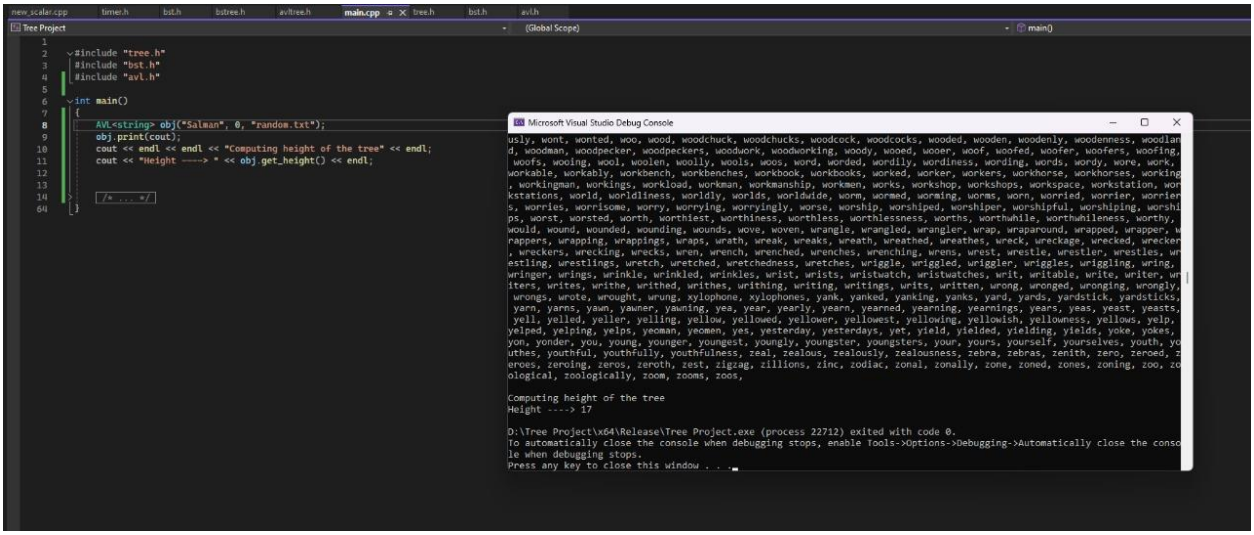
This indicates the **Power of a AVL Tree when data is sorted.**

ShortRandom.txt



For this purpose, AVL has a height of 4 with a total node count of 25. As data is random, tree doesn't grow on both sides. Instead, it performs rotations on its own and balance itself.

Random.txt



For this purpose, as data is random AVL has a height of 17 with a total node count of 45393. If we compare it to BST which had a height of 37 for the same Dataset, we can conclude that AVL is a far better approach for sorted and random data as it makes sure that our search operations are always in Log(N).

THE END.