

The experiment file is:

a b c d e f g h i j k l m n o p q r t s u v w x y z

This experimental file works because the elements inserted are in an increasing order. Therefore it requires balancing in the avl tree which rearranged to order from the normal BST. Therefore it can demonstrate the efficiency between the two trees clearly.

BST:

Left Links followed = 0

Right Links followed = 0

Total number of nodes = 26

Avg. node depth = 12.5

AVL:

Left Links followed = 0

Right Links followed = 0

Total number of nodes = 26

Single Rotations = 21

Double Rotations = 0

Avg. node depth = 3

As the data has shown, the average node depth for BST is much larger than that of AVL tree

In many cases, AVL tree's worst case find(),insert(),remove() method is always $O(\log(n))$ while BST's worst case find(),insert(),remove() method is rather $O(n)$ which shows that AVL tree is much more efficient.

However, due to the complicated nature of balancing method in AVL tree, it is very difficult to implement. As the example indicated, it takes a total of 21 single rotations to complete the balancing process. On the other hand, removing an element may also cause re-balancing the tree.