Abstract

AVL tree is the first dynamic tree in data structure which minimizes its height during insertion and deletion operations. This is because searching time is directly proportional to the height of binary search tree (BST) [1-9]. When insertion operation is performed it may result into increasing the height of the tree and when deletion is performed it may result into decreasing the height. To make the BST a height balance tree (AVL tree) creators of the AVL tree proposed various rotations. This paper proposes the balancing of the AVL tree using the concept of virtual node. This virtual node is a hypothetical node which is inserted into the inorder traversal of the BST and by doing the inorder traversal (left, root, right) we make a BST. Ultimately this virtual node is deleted to get an AVL tree.
Balancing of AVL tree using virtual node


Index Terms

Computer Science
Data Structures

Key words

AVL
BST
Insertion
Deletion
Virtual Node