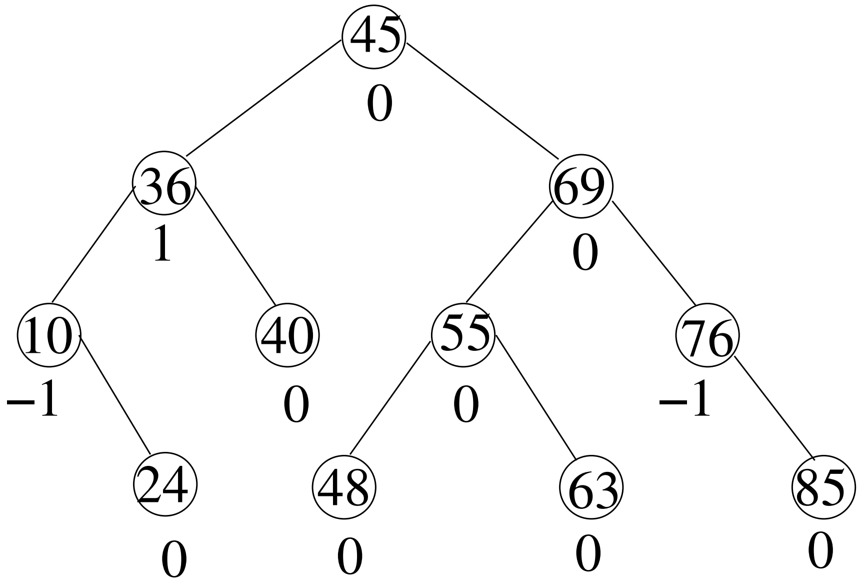
AVL tree

- BST which preserves **height balanced** property.
- At each internal node the difference in height of left and right subtrees is either 0, or 1, or -1.
- Example of an AVL tree.



AVL tree

- Since the tree is kept height balanced, searching can be accomplished by O($\log n$) time.
- Insertion or deletion may obviously disturb the height balance property.
- After every such updates we must ensure that the height balance property is preserved.
- The operation which restores the height balance property is called *rotation*.
- Rotation is an inorder preserving operation. Therefore, it does not disturb the search property.
- Rotation operation takes O(1) time.