

Introduction

The AVL tree is a self-balancing binary search tree where the difference in heights between the left and right subtrees of any node (known as the balance factor) is at most 1. When an imbalance occurs due to insertion or deletion, rotations are performed to restore balance.

Step-by-Step Process

- **Insertion:** Nodes are inserted following the binary search tree rules. After insertion, the height of each node is updated, and the balance factor is checked. If the balance factor of any node becomes greater than 1 or less than -1, rotations are performed to restore balance.
- **Rotations:** Depending on the type of imbalance (left-heavy or right-heavy), one or more rotations are performed. The types of rotations include:
 - **Right Rotation:** Applied when a left-heavy subtree needs balancing.
 - **Left Rotation:** Applied when a right-heavy subtree needs balancing.
 - **Left-Right Rotation:** Applied when a left subtree has a right-heavy child.
 - **Right-Left Rotation:** Applied when a right subtree has a left-heavy child.

Steps Visualization

The following figures illustrate the AVL tree at various stages of the insertion and balancing process:



Figure 1: Step 1

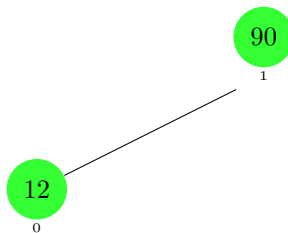


Figure 2: Step 2

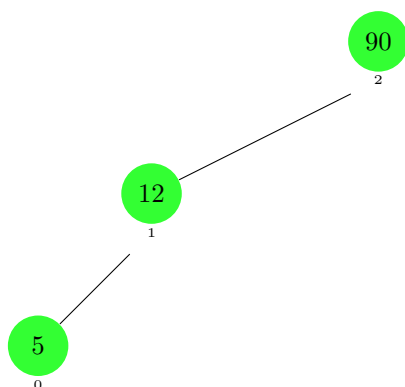


Figure 3: Step 3

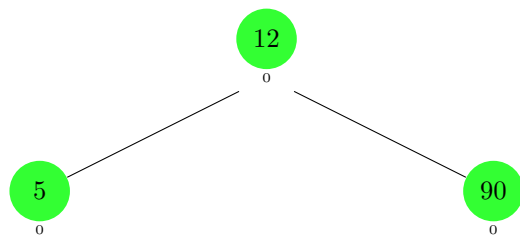


Figure 4: Step 4

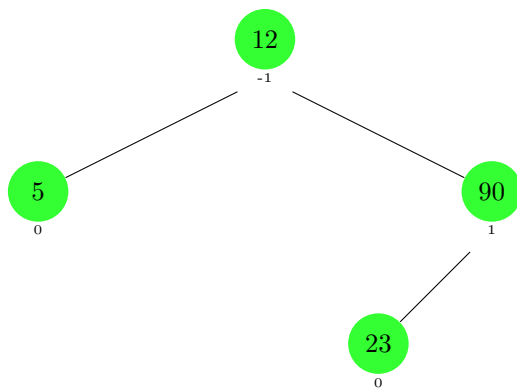


Figure 5: Step 5