Binary Tree Traversal Algorithms

In this document, we illustrate the process of traversing a binary tree using different traversal algorithms. The traversal methods covered are:

- **Pre-order Traversal:** In this traversal method, the algorithm visits the root node first, then recursively visits the left subtree, followed by the right subtree. This method is useful for scenarios where you need to process the root node before its children.
- **In-order Traversal:** This method visits the left subtree first, then the root node, and finally the right subtree. It is commonly used to retrieve the elements of the tree in sorted order, assuming the tree is a binary search tree (BST).
- **Post-order Traversal:** The algorithm first visits the left subtree, then the right subtree, and finally the root node. This traversal method is particularly useful for deleting nodes or evaluating expressions in expression trees.

Detailed Traversal Steps

The following figures illustrate the binary tree at various stages of traversal. Each figure shows the state of the tree with the current node being processed highlighted.

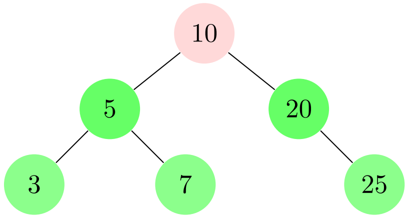

Figure 1: Initial Tree State

Traversal Steps

The following figures depict the binary tree as it is traversed using the selected traversal method. Each step highlights the node currently being processed.

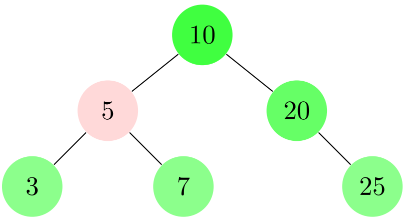

Figure 2: Step 2

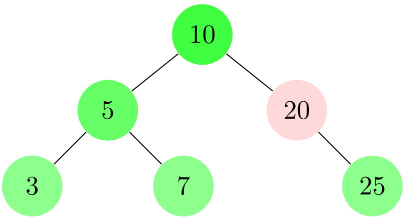

Figure 3: Step 3

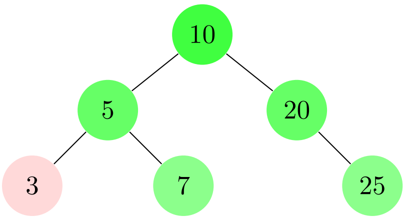

Figure 4: Step 4

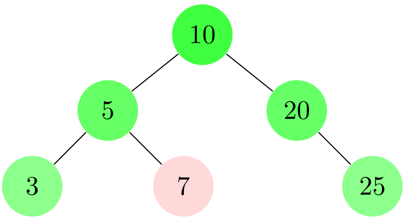

Figure 5: Step 5

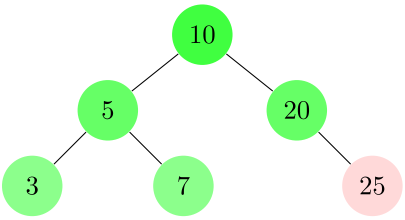

Figure 6: Step 6