Binary Tree Traversal Algorithms

Introduction

This document provides a visual representation of binary tree traversal algorithms. We will explore three different traversal methods: pre-order, in-order, and post-order. Each method is demonstrated through a series of figures that show the binary tree at various stages of traversal.

Traversal Algorithms

- **Pre-order Traversal:** In pre-order traversal, the nodes are recursively visited in the following order: root, left subtree, right subtree. This traversal method is useful for creating a prefix expression of the tree.
- **In-order Traversal:** In in-order traversal, the nodes are recursively visited in this order: left subtree, root, right subtree. This method is particularly useful for binary search trees (BSTs) as it retrieves the nodes in ascending order.
- **Post-order Traversal:** In post-order traversal, the nodes are recursively visited in the order: left subtree, right subtree, root. This traversal method is useful for deleting nodes or evaluating postfix expressions.

Initial Tree State

The following figure shows the initial state of the binary tree before any traversal has occurred. This provides a baseline for understanding how the tree changes during traversal.

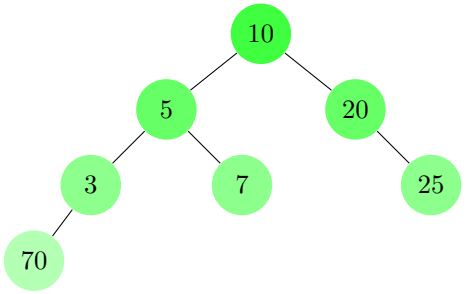

Figure 1: Initial Tree State

Traversal Steps

The following figures illustrate the binary tree at different stages of traversal, depending on the chosen traversal method. Each step shows the tree with the current node being processed highlighted.

Figure 2: Traversal Step 2

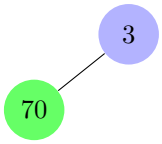

Figure 3: Traversal Step 3

Figure 4: Traversal Step 4

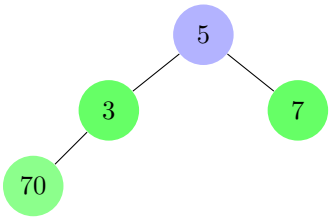

Figure 5: Traversal Step 5

Figure 6: Traversal Step 6

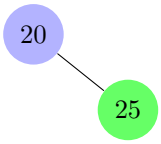

Figure 7: Traversal Step 7

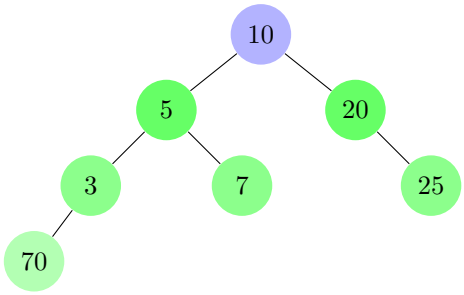

Figure 8: Traversal Step 8