

Advanced Concepts in Binary Trees

By Popa Ioan-Ciprian

1 Introduction

Binary trees are hierarchical data structures that organize data in a tree-like fashion. Each node in a binary tree has at most two children, known as the left child and the right child. Binary trees are extensively used in computer science due to their efficient storage and retrieval properties. In this document, we delve into advanced concepts and operations related to binary trees, accompanied by illustrative diagrams.

2 Basic Concepts

2.1 Definition

A binary tree is a hierarchical structure composed of nodes, where each node has at most two children: a left child and a right child. The topmost node is called the root.

2.2 Properties

- Every non-empty binary tree must have a root node.
- Each node in a binary tree can have at most two children.
- Binary trees can be empty, meaning they contain no nodes.
- The height of a binary tree is the maximum number of edges from the root node to any leaf node.

2.3 Types of Binary Trees

1. **Full Binary Tree:** Every non-leaf node has exactly two children.
2. **Complete Binary Tree:** All levels of the tree are completely filled except possibly for the last level, which is filled from left to right.
3. **Perfect Binary Tree:** All internal nodes have two children, and all leaf nodes are at the same level.

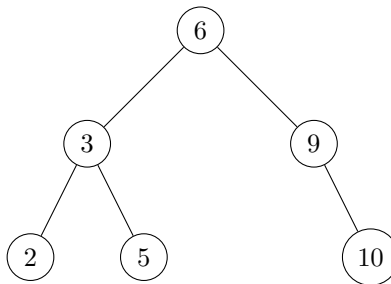
3 Operations

3.1 Insertion

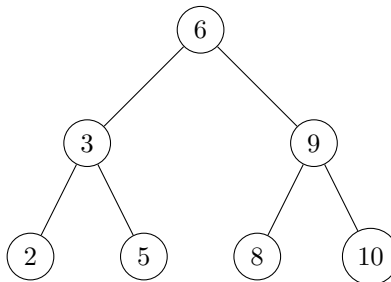
Binary tree insertion involves adding a new node to the tree while maintaining the binary tree properties. When inserting an element into a binary search tree, the element is compared with the value of each node starting from the root. If the element is less than the current node's value, it is inserted into the left subtree; otherwise, it is inserted into the right subtree. The process continues recursively until a suitable position for insertion is found.

3.1.1 Example

Consider inserting a new node with value 8 into the following binary search tree:



After insertion:



3.2 Traversal

Traversal is the process of visiting all the nodes in a tree in a specific order.

3.2.1 Inorder Traversal

In inorder traversal, nodes are visited in the order: left subtree, root, right subtree.

3.2.2 Preorder Traversal

In preorder traversal, nodes are visited in the order: root, left subtree, right subtree.

3.2.3 Postorder Traversal

In postorder traversal, nodes are visited in the order: left subtree, right subtree, root.

4 Applications

Binary trees have diverse applications, including:

- Binary search trees for efficient searching, insertion, and deletion.
- Expression trees for representing mathematical expressions.
- Huffman trees for data compression.
- Decision trees for classification and decision-making processes.

5 Conclusion

Binary trees are versatile data structures with a wide range of applications. Understanding their operations and properties is essential for designing efficient algorithms and solving complex problems.