**Beginner Level (Understanding Tree Basics)**

1. **Binary Tree Inorder Traversal**
   * **Learn basic traversal techniques (recursive and iterative).**
   * **Tags: Easy.**
2. **Binary Tree Preorder Traversal**
   * **Preorder traversal of a binary tree.**
   * **Tags: Easy.**
3. **Binary Tree Postorder Traversal**
   * **Postorder traversal of a binary tree.**
   * **Tags: Easy.**
4. **Maximum Depth of Binary Tree**
   * **Find the maximum depth of a tree using recursion.**
   * **Tags: Easy.**
5. **Same Tree**
   * **Check if two binary trees are identical.**
   * **Tags: Easy.**
6. **Symmetric Tree**
   * **Determine if a binary tree is symmetric.**
   * **Tags: Easy, recursion.**
7. **Invert Binary Tree**
   * **Mirror a binary tree.**
   * **Tags: Easy.**
8. **Path Sum**
   * **Check if there’s a root-to-leaf path with a specific sum.**
   * **Tags: Easy.**
9. **Binary Tree Level Order Traversal**
   * **Breadth-first traversal using a queue.**
   * **Tags: Medium.**
10. **Binary Tree Zigzag Level Order Traversal**
    * **Level-order traversal with alternating directions.**
    * **Tags: Medium.**

**Intermediate Level (Tree Manipulation and Properties)**

1. **Diameter of Binary Tree**
   * **Find the longest path between two nodes.**
   * **Tags: Easy.**
2. **Lowest Common Ancestor of a Binary Search Tree**
   * **Simple recursive LCA for a BST.**
   * **Tags: Easy.**
3. **Lowest Common Ancestor of a Binary Tree**
   * **Recursive LCA for any binary tree.**
   * **Tags: Medium.**
4. **Construct Binary Tree from Preorder and Inorder Traversal**
   * **Reconstruct a tree using traversal orders.**
   * **Tags: Medium.**
5. **Validate Binary Search Tree**
   * **Check if a binary tree is a valid BST.**
   * **Tags: Medium.**
6. **Binary Tree Right Side View**
   * **Return the nodes visible from the right side.**
   * **Tags: Medium.**
7. **Populating Next Right Pointers in Each Node**
   * **Link tree nodes at the same level.**
   * **Tags: Medium.**
8. **Kth Smallest Element in a BST**
   * **Use in-order traversal for kth smallest.**
   * **Tags: Medium.**
9. **Balanced Binary Tree**
   * **Check if a binary tree is height-balanced.**
   * **Tags: Easy.**
10. **Serialize and Deserialize Binary Tree**
    * **Implement serialization and deserialization of a binary tree.**
    * **Tags: Hard.**

**Advanced Level (Complex Concepts and Techniques)**

1. **Flatten Binary Tree to Linked List**
   * **Convert a binary tree into a flattened linked list.**
   * **Tags: Medium.**
2. **Recover Binary Search Tree**
   * **Recover a BST after two nodes are swapped.**
   * **Tags: Medium.**
3. **Count Complete Tree Nodes**
   * **Count nodes efficiently in a complete binary tree.**
   * **Tags: Medium.**
4. **Path Sum II**
   * **Find all root-to-leaf paths that sum to a target.**
   * **Tags: Medium.**
5. **Sum Root to Leaf Numbers**
   * **Calculate the sum of all root-to-leaf numbers.**
   * **Tags: Medium.**
6. **House Robber III**
   * **Solve the house robber problem on a binary tree using DP.**
   * **Tags: Medium.**
7. **Binary Tree Maximum Path Sum**
   * **Find the maximum path sum in a tree.**
   * **Tags: Hard.**
8. **Construct Binary Tree from Inorder and Postorder Traversal**
   * **Build a tree using traversal orders.**
   * **Tags: Medium.**
9. **Sum of Distances in Tree**
   * **Dynamic programming on trees to calculate node distances.**
   * **Tags: Hard.**
10. **All Nodes Distance K in Binary Tree**
    * **Find all nodes at distance k from a target node.**
    * **Tags: Medium.**

**Expert Level (Graph Theory, Advanced Tree Concepts)**

1. **Binary Tree Cameras**
   * **Place cameras to monitor all nodes with minimal count.**
   * **Tags: Hard, greedy.**
2. **Delete Node in a BST**
   * **Delete a node in a BST while maintaining structure.**
   * **Tags: Medium.**
3. **Construct Quad Tree**
   * **Build a quad-tree from a grid.**
   * **Tags: Medium.**
4. **Vertical Order Traversal of a Binary Tree**
   * **Traverse nodes vertically by columns.**
   * **Tags: Hard.**
5. **Split BST**
   * **Split a BST into two trees based on a target value.**
   * **Tags: Hard.**
6. **Closest Binary Search Tree Value II**
   * **Find k closest values to a target in a BST.**
   * **Tags: Hard.**
7. **Find Duplicate Subtrees**
   * **Identify duplicate subtrees in a binary tree.**
   * **Tags: Medium.**
8. **Binary Tree Longest Consecutive Sequence**
   * **Find the longest consecutive sequence in a binary tree.**
   * **Tags: Medium.**
9. **Smallest Subtree with All the Deepest Nodes**
   * **Find the subtree containing all deepest nodes.**
   * **Tags: Medium.**
10. **Redundant Connection**
    * **Detect cycles in a graph representation of a tree.**
    * **Tags: Hard.**

**How to Progress**

1. **Start with Basic Traversals:**
   * **Problems 1–10 focus on understanding the structure and traversals of binary trees.**
2. **Learn Recursive and Iterative Approaches:**
   * **Solve problems 11–20 to master recursion and iterative solutions.**
3. **Dive Into Manipulations and Construction:**
   * **Problems 21–30 focus on manipulating tree structures and dynamic programming.**
4. **Master Advanced Techniques:**
   * **Problems 31–40 introduce optimization techniques, graph-theory-based tree problems, and edge cases.**