**Beginner Level:**

1. Implement a binary tree
2. Implement a binary search tree (BST)
3. Insert a node in a binary tree
4. Insert a node in a binary search tree
5. Delete a node from a binary search tree
6. Find the minimum value in a binary search tree
7. Find the maximum value in a binary search tree
8. Perform an inorder traversal of a binary tree
9. Perform a preorder traversal of a binary tree
10. Perform a postorder traversal of a binary tree
11. Perform a level order traversal of a binary tree
12. Calculate the height of a binary tree
13. Count the number of nodes in a binary tree
14. Count the number of leaves in a binary tree
15. Find the diameter of a binary tree
16. Check if a binary tree is balanced
17. Check if two binary trees are identical
18. Find the lowest common ancestor (LCA) of two nodes in a binary tree
19. Print all paths from root to leaf in a binary tree
20. Check if a binary tree is a binary search tree (BST)

**Medium Level:**

1. Construct a binary tree from inorder and preorder traversals
2. Construct a binary tree from inorder and postorder traversals
3. Construct a binary search tree from a sorted array
4. Find the kth smallest element in a binary search tree
5. Find the kth largest element in a binary search tree
6. Convert a binary tree to a doubly linked list
7. Convert a binary search tree to a doubly linked list
8. Flatten a binary tree to a linked list
9. Find the vertical order traversal of a binary tree
10. Find the top view of a binary tree
11. Find the bottom view of a binary tree
12. Find the right view of a binary tree
13. Find the left view of a binary tree
14. Find the zigzag level order traversal of a binary tree
15. Find the maximum path sum in a binary tree
16. Find the sum of all nodes in a binary tree
17. Find the sum of all left leaves in a binary tree
18. Find the sum of all right leaves in a binary tree
19. Find the sum of all nodes at the maximum depth of a binary tree
20. Check if a binary tree has a path sum equal to a given value

**Advanced Level:**

1. Implement an AVL tree (self-balancing binary search tree)
2. Implement a Red-Black tree
3. Implement a Splay tree
4. Implement a B-tree
5. Implement a Segment tree for range sum queries
6. Implement a Segment tree for range minimum queries
7. Implement a Fenwick tree (Binary Indexed Tree) for range sum queries
8. Find the lowest common ancestor (LCA) in a binary tree using Tarjan's offline algorithm
9. Convert a binary search tree to a balanced binary search tree
10. Implement the Cartesian tree and perform tree operations on it