

# CP-CS1-M

---

0. Implement traversals recursively - Inorder, Preorder, PostOrder, **LevelOrder**, 1. Implement traversals iteratively - Inorder, Preorder, **PostOrder, LevelOrder**

2. Print Left/**Right/Bottom**/Top view of the Binary Tree

3. **Construct tree from inorder and preorder traversal**

**(Easy to Medium)**

4. LCA of Binary Tree (Recursive/Iterative)

5. Diameter of Binary Tree

6. Sum of all nodes of Binary Tree (Easy)

7. Max Sum path from the leaf to leaf.

**8. Mirror Tree / Identical tree (Easy)**

9. Height of Binary Tree

10. Check if the tree is a (full binary tree/balanced binary tree/perfect

binary tree) or not 11. Serialize/Deserialize Binary Tree

**12. Connect Nodes on the same level (Hard)**

13. Convert each level in Binary Tree to Doubly LinkedList (Hard)

14. Reverse Level Order, Spiral Level Order, Boundary Traversal,

**Vertical Traversal** 15. [Construct Special Binary Tree from given](#)

[Inorder traversal](#)

**16. Print root to leaf path in Binary tree (Easy)**

17. Print Cousins of a given Nodes in a binary tree

**18. Print all nodes at K distance. (Hard)**

**19. Find Largest Subtree sum in Binary Tree (Easy to Medium)**

**H/W : Construct tree from inorder and postorder traversal**

**(Easy to Medium)**