Leetcode

Leetcode Solution 1:-

```
class Solution {
    private int val;
    private int depth;

    public TreeNode addOneRow(TreeNode root, int val, int depth) {
        if (depth == 1) {
            return new TreeNode(val, root, null);
        }
        this.val = val;
        this.depth = depth;
        dfs(root, 1);
        return root;
    }

    private void dfs(TreeNode root, int d) {
        if (root == null) {
            return;
        }
        if (d == depth - 1) {
            TreeNode l = new TreeNode(val, root.left, null);
            TreeNode r = new TreeNode(val, null, root.right);
            root.left = l;
            root.right = r;
            return;
        }
        dfs(root.left, d + 1);
        dfs(root.right, d + 1);
    }
}
```

Leetcode Solution 2:-

```
class Solution {
    public TreeNode pruneTree(TreeNode root) {
        if (root == null) {
            return null;
        }
        root.left = pruneTree(root.left);
        root.right = pruneTree(root.right);
        if (root.val == 0 && root.left == null && root.right == null) {
            return null;
        }
        return root;
    }
}
```