Leetcode

Leetcode Solution 1:-

```
class Solution {
    public String tree2str(TreeNode root) {
        if (root == null) {
            return "";
        }
        if (root.left == null && root.right == null) {
            return root.val + "";
        }
        if (root.right == null) {
            return root.val + "(" + tree2str(root.left) + ")";
        }
        return root.val + "(" + tree2str(root.left) + ")(" + tree2str(root.right) + ")";
    }
}
```

Leetcode Solution 2:-

```
class Solution {
    public List<List<String>> findDuplicate(String[] paths) {
        Map<String, List<String>> d = new HashMap<>();
        for (String p : paths) {
            String[] ps = p.split(" ");
            for (int i = 1; i < ps.length; ++i) {
                int j = ps[i].indexOf('(');
                String content = ps[i].substring(j + 1, ps[i].length() - 1);
                String name = ps[0] + '/' + ps[i].substring(0, j);
                d.computeIfAbsent(content, k -> new
ArrayList<>()).add(name);
            }
        }
        List<List<String>> ans = new ArrayList<>();
        for (var e : d.values()) {
            if (e.size() > 1) {
                ans.add(e);
            }
        }
        return ans;
    }
}
```