652. Find Duplicate Subtrees

- Difficulty:Medium
- Total Accepted:2.3K
- Total Submissions:8.3K
- Contributor:Stomach_ache
-
- **Subscribe** to see which companies asked this question.

Related Topics

Tree

Similar Questions

Serialize and Deserialize Binary TreeSerialize and Deserialize BSTConstruct String from Binary Tree

Given a binary tree, return all duplicate subtrees. For each kind of duplicate subtrees, you only need to return the root node of any **one**of them.

Two trees are duplicate if they have the same structure with same node values.

Example 1:

```
        1
       / \
      2   3
     /   / \
    4   2   4
       /
      4
```

The following are two duplicate subtrees:

```
      2
     /
    4
```

and

```
    4
```

Therefore, you need to return above trees' root in the form of a list.

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    string serialize(TreeNode* root,unordered_map<string,int> &map,
vector<TreeNode*> &res)
    {
        if(root==NULL) return "";
        string res_str = "(" + serialize(root->left,map,res) +
","+to_string(root->val) + ","+serialize(root->right,map,res)+")";
        if(map[res_str]==1) res.push_back(root);
        map[res_str] =  map[res_str]+1;
        return res_str;
    }

    vector<TreeNode*> findDuplicateSubtrees(TreeNode* root) {
        unordered_map<string,int> map;
        vector<TreeNode*> res;
        serialize(root,map,res);
        return res;
    }
};
```