102. Binary Tree Level Order Traversal ★

Question | Editorial Solution

My Submissions (/problems/binary-tree-level-order-traversal/submissions/)

Total Accepted: **127510** Total Submissions: **360632** Difficulty: **Easy**

Notes

Given a binary tree, return the *level order* traversal of its nodes' values. (ie, from left to right, level by level).

For example:

Given binary tree `[3,9,20,null,null,15,7]`,

```
    3
   / \
  9  20
    /  \
   15   7
```

return its level order traversal as:

```
[
  [3],
  [9,20],
  [15,7]
]
```

Subscribe (/subscribe/) to see which companies asked this question

Show Tags

Show Similar Problems

Have you met this question in a real interview? Yes No

Discuss (https://leetcode.com/discuss/questions/oj/binary-tree-level-order-traversal)

Pick One (/problems/random-one-question/)

C++

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
```

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

```cpp
class Solution {
public:
    vector<vector<int>> levelOrder(TreeNode* root) {
        std::queue<TreeNode*> s1, s2;
        std::queue<TreeNode*> *ptr1 = &s1, *ptr2 = &s2;
        vector<vector<int>> result;
        if (!root) return result;
        s1.push(root);
        while (!ptr1->empty()) {
            result.push_back(vector<int>());
            while (!ptr1->empty()) {
                TreeNode* now = ptr1->front();
                ptr1->pop();
                result.back().push_back(now->val);
                if (now->left) ptr2->push(now->left);
                if (now->right) ptr2->push(now->right);
            }
            swap(ptr1, ptr2);
        }
        return result;
    }
};
```

Notes

Custom Testcase

Run Code | Submit Solution