

# 102. Binary Tree Level Order Traversal

## 102 Binary Tree Level Order Traversal

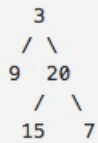
- Breadth-first Search + Queue + Tree

### Description

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] ,



return its level order traversal as:

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

### 1. Thought line

### 2. Breadth-first Search + Queue + Tree

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

class Solution {
public:
    vector<vector<int>> levelOrder(TreeNode* root) {
        vector<vector<int>> result;
        queue<TreeNode*> que;
        if (root!=nullptr) que.emplace(root);

        while (!que.empty() || que.front()!=nullptr){
            queue<TreeNode*> tempQue;
            vector<int> tempVec;
            while (!que.empty()){
                tempVec.push_back(que.front()->val);
                if (que.front()->left!=nullptr ) tempQue.push(que.front()->left);
            }
        }
    }
};
```

```
};  
    if (que.front()->right!=nullptr) tempQue.push(que.front()->right);  
    que.pop();  
}  
if(!tempVec.empty()) result.push_back(tempVec);  
else break;  
if(!tempQue.empty()) que.swap(tempQue);  
else break;  
}  
return result;  
};
```