

102. Binary Tree Level Order Traversal

102 Binary Tree Level Order Traversal

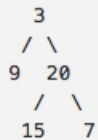
- Breadth-first Search + Queue + Tree

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

```
1 /**
2  * Definition for a binary tree node.
3  * struct TreeNode {
4  *     int val;
5  *     TreeNode *left;
6  *     TreeNode *right;
7  *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
8  * };
9  */
10 #include <queue>
11
12 class Solution {
13 public:
14     vector<vector<int>> levelOrder(TreeNode* root) {
15         vector<vector<int>> result;
16         queue<TreeNode*> que;
17         if (root!=nullptr) que.emplace(root);
18
19         while (!que.empty() || que.front()!=nullptr){
20             queue<TreeNode*> tempQue;
21             vector<int> tempVec;
22             while (!que.empty()){
23                 tempVec.push_back(que.front()->val);
24                 if (que.front()->left!=nullptr ) tempQue.push(que.front()->left);
25                 if (que.front()->right!=nullptr) tempQue.push(que.front()->right);
26                 que.pop();
27             }
28             if(!tempVec.empty()) result.push_back(tempVec);
29             else break;
```

```
30         if(!tempQue.empty()) que.swap(tempQue);
31         else break;
32
33     }
34     return result;
35 }
36
37 };
```