

二叉树的层序遍历

给你一个二叉树，请你返回其按 **层序遍历** 得到的节点值。（即逐层地，从左到右访问所有节点）。

示例：

二叉树：[3,9,20,null,null,15,7],



返回其层序遍历结果：

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

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left
 * ), right(right) {}
 * };
 */
class Solution {
public:
    vector<vector<int>> levelOrder(TreeNode* root) {
        queue<TreeNode*> q;
        vector<vector<int>> mat;
        if(root)
```

```
{
    q.push(root);
}
while(!q.empty())
{
    int rowsize=q.size();//获取每一层节点个数
    vector<int> tmp;//存储每一行数据
    while(rowsize--)
    {
        TreeNode* cur=q.front();
        q.pop();
        tmp.push_back(cur->val);
        //接下来判断是否有左右节点
        if(cur->left)
        {
            q.push(cur->left);
        }
        if(cur->right)
        {
            q.push(cur->right);
        }
    }
    mat.push_back(tmp);
}
return mat;
}
};
```