

二叉树的遍历方法

在面试中，遇到二叉树需要遍历的问题，尽量不要用递归，考官希望你用迭代去解决这个问题。我曾经在面试中多次遇到二叉树遍历的问题，考官都强调要用迭代的做法，因此二叉树的前序，中序，后序，层序的迭代遍历很重要。

先定义一棵树：

```
public class TreeNode{  
    int val;  
  
    TreeNode left;  
  
    TreeNode right;  
  
    TreeNode(int val, TreeNode left, TreeNode right){  
  
        this.val = val;  
  
        this.left = left;  
  
        this.right = right;  
  
    }  
}
```

-----前序遍历-----

```
class Solution {  
    public List<Integer> preorderTraversal(TreeNode root) {  
        List<Integer> res = new ArrayList<>();  
        TreeNode node = root;  
        Stack<TreeNode> stack = new Stack<>();  
        if(root == null) return res;  
        while(!res.isEmpty() || node != null){  
            while(node != null){  
                res.add(node.val);  
                stack.push(node);  
                node = node.left;  
            }  
            node = stack.pop();  
        }  
    }  
}
```

```

        node = node.right;
    }
    return res;
}
}

```

-----中序-----

```

class Solution {
public List<Integer> preorderTraversal(TreeNode root) {
    List<Integer> res = new ArrayList<>();
    TreeNode node = root;
    Stack<TreeNode>() stack = new Stack<>();
    if(root == null) return res;
    while(!res.isEmpty() || node != null){
        while(node != null){
            stack.push(node);
            node = node.left;
        }
        node = stack.pop();
        res.add(node.val);
        node = node.right;
    }
    return res;
}
}

```

-----后序-----

```

class Solution {
public List<Integer> preorderTraversal(TreeNode root) {
    LinkedList<Integer> res = new LinkedList<>();
    TreeNode node = root;
    Stack<TreeNode>() stack = new Stack<>();
    if(root == null) return res;
    while(!res.isEmpty() || node != null){
        while(node != null){
            stack.push(node);
            res.addFirst(node.val);
            node = node.right;
        }
        node = stack.pop();
        node = node.left;
    }
    return res;
}
}

```

102.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.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode(int x) { val = x; }
 * }
 */
class Solution {
public List<List<Integer>> levelOrder(TreeNode root) {
    List<List<Integer>> res = new ArrayList<List<Integer>>();
    Queue<TreeNode> quene = new LinkedList<TreeNode>();
```

```
if(root == null)
return res;
    quene.offer(root);
while(!quene.isEmpty()){
int count = quene.size();
    List<Integer> list = new ArrayList<Integer>();
for(int i = 1; i <= count; ++i){
    TreeNode temp = quene.poll();
    list.add(temp.val);
if(temp.left != null){
    quene.offer(temp.left);
}
if(temp.right != null){
    quene.offer(temp.right);
}
}
    res.add(list);
}
return res;
}
```