

面试题32： 从上往下打印二叉树

题目描述

从上往下打印出二叉树的每个节点，同层节点从左至右打印。

解题思路

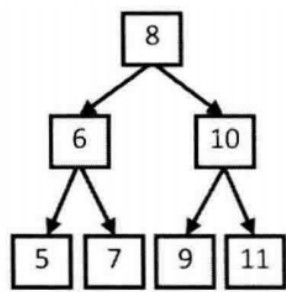


图 4.6 一棵二叉树，从上到下按层打印的顺序为 8,6,10,5,7,9,11

表 4.4 按层打印图 4.6 中的二叉树的过程

步 骤	操 作	队 列
1	打印节点 8	节点 6、节点 10
2	打印节点 6	节点 10、节点 5、节点 7
3	打印节点 10	节点 5、节点 7、节点 9、节点 11
4	打印节点 5	节点 7、节点 9、节点 11
5	打印节点 7	节点 9、节点 11
6	打印节点 9	节点 11
7	打印节点 11	

通过上面具体例子的分析，我们可以找到从上到下打印二叉树的规律：
每次打印一个节点的时候，如果该节点有子节点，则把该节点的子节点放到一个队列的末尾。接下来到队列的头部取出最早进入队列的节点，重复前面的打印操作，直至队列中所有的节点都被打印出来。

C++

```
1  /*
2  struct TreeNode {
3      int val;
4      struct TreeNode *left;
5      struct TreeNode *right;
6      TreeNode(int x) :
7          val(x), left(NULL), right(NULL)
8      {
9      }
10 };*/
11 class Solution {
12 public:
13     vector<int>
14     PrintFromTopToBottom(TreeNode* root) {
15         TreeNode* fr;
16         if(root == nullptr)
17             return result;
18         que.push(root);
19         while(!que.empty()){
20             fr = que.front();
21             result.push_back(fr -> val);
22             if(fr -> left != nullptr)
23                 que.push(fr -> left);
24             if(fr -> right != nullptr)
25                 que.push(fr -> right);
26             que.pop();
27         }
28     }
29 }
```

```

25         }
26         return result;
27     }
28 private:
29     vector<int> result;
30     queue<TreeNode*> que;
31 };

```

Java

```

1  import java.util.ArrayList;
2  /**
3  public class TreeNode {
4      int val = 0;
5      TreeNode left = null;
6      TreeNode right = null;
7
8      public TreeNode(int val) {
9          this.val = val;
10
11     }
12
13 }
14 */
15 public class Solution {
16     public ArrayList<Integer>
PrintFromTopToBottom(TreeNode root) {

```

```

17         ArrayList<Integer> list = new
ArrayList<>();
18         ArrayList<TreeNode> queue = new
ArrayList<>();
19         if(root == null)
20             return list;
21         queue.add(root);
22         while(queue.size() != 0){
23             TreeNode temp = queue.remove(0);
24             if(temp.left != null){
25                 queue.add(temp.left);
26             }
27             if(temp.right != null){
28                 queue.add(temp.right);
29             }
30             list.add(temp.val);
31         }
32         return list;
33     }
34 }

```

Python 2.7.3

```

1  # -*- coding:utf-8 -*-
2  # class TreeNode:
3  #     def __init__(self, x):
4  #         self.val = x
5  #         self.left = None

```

```
6 #         self.right = None
7 class Solution:
8     # 返回从上到下每个节点值列表，例：[1,2,3]
9     def PrintFromTopToBottom(self, root):
10         # write code here
11         if not root:
12             return []
13         result = []
14         tmp = [root]
15         while len(tmp):
16             cur = tmp.pop(0)
17             result.append(cur.val)
18             if cur.left:
19                 tmp.append(cur.left)
20             if cur.right:
21                 tmp.append(cur.right)
22         return result
```