

剑指Offer（六十）：把二叉树打印成多行

🕒 2018年1月29日 11:38:22 🗂 2 🌡 3,098 °C 🛠 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

从上到下按层打印二叉树，同一层结点从左至右输出。每一层输出一行。

1、思路

这道题比上一道题《剑指Offer（五十九）：按之字顺序打印二叉树》简单一些，牛客网将这两道题应该是放错顺序了。

思路和上一道题一样，区别在于，这把是先入先出，使用队列即可。

2、代码

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<vector<int> > Print(TreeNode* pRoot) {
14         vector<vector<int> > result;
15         if(pRoot == NULL){
16             return result;
17         }
18         queue<TreeNode* > nodes[2];
19         nodes[0].push(pRoot);
20         while(!nodes[0].empty() || !nodes[1].empty()){
21             vector<int> v[2];
22             while(!nodes[0].empty()){
23                 v[0].push_back(nodes[0].front()->val);
24                 if(nodes[0].front()->left != NULL){
25                     nodes[1].push(nodes[0].front()->left);
26                 }
27                 if(nodes[0].front()->right != NULL){
28                     nodes[1].push(nodes[0].front()->right);
29                 }
30                 nodes[0].pop();
31             }
32             if(!v[0].empty()){
33                 result.push_back(v[0]);
34             }
35             while(!nodes[1].empty()){
36                 v[1].push_back(nodes[1].front()->val);
37                 if(nodes[1].front()->left != NULL){
38                     nodes[0].push(nodes[1].front()->left);
39                 }
40             }
41         }
42     }
43 }
```

```
40         if(nodes[1].front()->right != NULL){
41             nodes[0].push(nodes[1].front()->right);
42         }
43         nodes[1].pop();
44     }
45     if(!v[1].empty()){
46         result.push_back(v[1]);
47     }
48 }
49 return result;
50 }
51 };
```



微信公众号

分享技术，乐享生活：微信公众号搜索

「JackCui-AI」关注一个在互联网摸爬滚  
打的潜行者。

简洁是智慧的灵魂，冗长是肤浅的藻饰。--- 莎士比亚