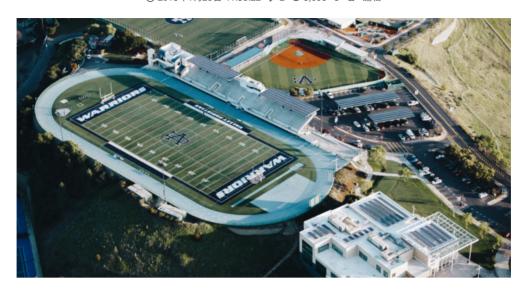
剑指Offer(六十): 把二叉树打印成多行



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++:

```
C+
2
3
4
5
6
7
8
9
10
       struct TreeNode {
                int val;
struct TreeNode *left;
struct TreeNode *right;
                 TreeNode(int x)
                                   val(x), left(NULL), right(NULL) {
11 class Solution {
                         vector<vector<int> > Print(TreeNode* pRoot) {
   vector<vector<int> > result;
   if(pRoot == NULL){
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
                                            return result
                                   queue<TreeNode* > nodes[2];
                                   queue<freeNode > nodes[2],
nodes[0].push(pRoot);
while(!nodes[0].empty() || !nodes[1].empty()){
                                            tet(indes[0].empty()){
vector<int> v[2];
while(!nodes[0].empty()){
    v[0].push_back(nodes[0].front()->val);
    if(nodes[0].front()->left != NULL){
        nodes[1].push(nodes[0].front()->left);
}
                                                     if(nodes[0].front()->right != NULL){
    nodes[1].push(nodes[0].front()->right);
                                                     nodes[0].pop();
                                            if(!v[0].empty()){
    result.push_back(v[0]);
                                            }
while(!nodes[1].empty()){
  v[1].push_back(nodes[1].front()->val);
  if(nodes[1].front()->left != NULL){
      nodes[0].push(nodes[1].front()->left);
}
```



微信公众号

分享技术,乐享生活:微信公众号搜索「JackCui-Al」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