```
题目：分行打印二叉树
/*
struct TreeNode {
    int val;
    struct TreeNode *left;
    struct TreeNode *right;
    TreeNode(int x) :
            val(x), left(NULL), right(NULL) {
    }
};
*/
class Solution {
public:
        vector<vector<int> > Print(TreeNode* pRoot) {

            // 存储结果
            vector<vector<int>>  result; // 存储全部节点
            vector<int> temp;            // 存储某一层节点

            // 边界条件
            if(pRoot == nullptr)
                return result;

            // 辅助容器：队列
            queue<TreeNode*> q;
            TreeNode* fr;
            int now_level=1;
            int next_level=0;

            // 根节点入队列
            q.push(pRoot);

            // 遍历队列
            while(!q.empty())
            {
                // 节点弹出队列
                fr=q.front();
                temp.push_back(fr->val);
                q.pop();

                // 遍历节点左右子树
                if(fr->left != NULL){
                    q.push(fr->left);
                    ++next_level;
                }
                if(fr->right != NULL){
                    q.push(fr->right);
                    ++next_level;
                }
                --now_level;

                // 判断当前层是否打印完
                if(now_level==0){
                    now_level=next_level;
                    next_level=0;
                    result.push_back(temp);
                    temp.clear(); // 清除一维vector
                }
            }

           return result;
        }
};
```