/\*

Given a binary tree, imagine yourself standing on the right side of it, return the values of the nodes you can see ordered from top to bottom.

For example:

Given the following binary tree,

1 <---

/ \

2 3 <---

\ \

5 4 <---

You should return [1, 3, 4].

层数遍历

中左右的顺序

\*/

/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode(int x) : val(x), left(NULL), right(NULL) {}

\* };

\*/

class Solution {

public:

void guocheng(vector<int> & ret,TreeNode\* root,int n)

{

if(!root)

return;

else

{

if(n==ret.size())

ret.push\_back(root->val);

else

ret[n]=root->val;

}

guocheng(ret,root->left,n+1);

guocheng(ret,root->right,n+1);

}

vector<int> rightSideView(TreeNode\* root)

{

vector<int> ret;

guocheng(ret,root,0);

return ret;

}

};