

```

1  /*
2  给定一棵二叉树，想象自己站在它的右侧，按照从顶部到底部的顺序，返回从右侧所能看到的节点值。
3
4  示例：
5
6  输入：[1,2,3,null,5,null,4]
7  输出：[1, 3, 4]
8  解释：
9
10     1             <---
11    /  \
12   2    3         <---
13    \    \
14     5    4         <---
15
16  来源：力扣（LeetCode）
17  链接：https://leetcode-cn.com/problems/binary-tree-right-side-view
18  著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
19  */

```

分析:

使用BFS,保存每一层的最后一个元素即可.

方法一:C++\_BFS

```

1  /**
2   * Definition for a binary tree node.
3   * struct TreeNode {
4   *     int val;
5   *     TreeNode *left;
6   *     TreeNode *right;
7   *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
8   * };
9  */
10 class Solution
11 {
12 public:
13     vector<int> rightSideview(TreeNode* root)
14     {
15         vector<int>      ret_vi      ;
16         queue<TreeNode*>  qtn        ;
17         TreeNode*       temp       = NULL ;
18         int              num_1      = 0   ;
19         int              num_2      = 0   ;
20         int              i          = 0   ;
21         if(root)
22         {
23             qtn.push(root);
24             num_2 = 1;
25             num_1 = 0;
26             while(num_2)
27             {

```

```
28         num_1 = num_2;
29         num_2 = 0;
30         for(i=0;i<num_1;i++)
31         {
32             temp = qtn.front();
33             qtn.pop();
34             if(temp->left)
35             {
36                 qtn.push(temp->left);
37                 num_2++;
38             }
39             if(temp->right)
40             {
41                 qtn.push(temp->right);
42                 num_2++;
43             }
44         }
45         ret_vi.push_back(temp->val);
46     }
47 }
48 return ret_vi;
49 }
50 };
51 /*
52 执行结果:
53 通过
54 显示详情
55 执行用时 :4 ms, 在所有 cpp 提交中击败了91.62%的用户
56 内存消耗 :9.6 MB, 在所有 cpp 提交中击败了41.64%的用户
57 */
```