

0199. 二叉树的右视图

👤 ITCharge 🕒 大约 1 分钟

- 标签：树、深度优先搜索、广度优先搜索、二叉树
- 难度：中等

题目链接

- [0199. 二叉树的右视图 - 力扣](#)

题目大意

描述：给定一棵二叉树的根节点 `root` 。

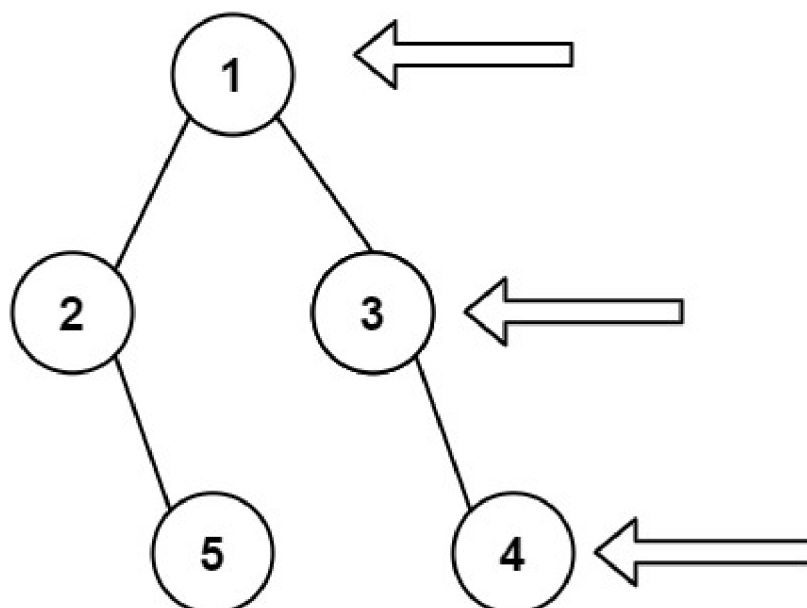
要求：按照从顶部到底部的顺序，返回从右侧能看到的节点值。

说明：

- 二叉树的节点个数的范围是 $[0, 100]$
- $-100 \leq Node.val \leq 100$ 。

示例：

- 示例 1：



py

输入: [1,2,3,null,5,null,4]

输出: [1,3,4]

- 示例 2:

py

输入: [1,null,3]

输出: [1,3]

解题思路

思路 1: 广度优先搜索

使用广度优先搜索对二叉树进行层次遍历。在遍历每层节点的时候, 只需要将最后一个节点加入结果数组即可。

思路 1: 代码

py

```
class Solution:
    def rightSideView(self, root: TreeNode) -> List[int]:
        if not root:
            return []
        queue = [root]
        order = []
        while queue:
            size = len(queue)
            for i in range(size):
                curr = queue.pop(0)
                if curr.left:
                    queue.append(curr.left)
                if curr.right:
                    queue.append(curr.right)
            if i == size - 1:
                order.append(curr.val)
        return order
```

思路 1：复杂度分析

- **时间复杂度：** $O(n)$ ，其中 n 是二叉树的节点数目。
- **空间复杂度：** $O(n)$ 。递归函数需要用到栈空间，栈空间取决于递归深度，最坏情况下递归深度为 n ，所以空间复杂度为 $O(n)$ 。