# 题目

给你一个二叉搜索树的根节点root，和一个由正整数组成、长度为n的数组queries。

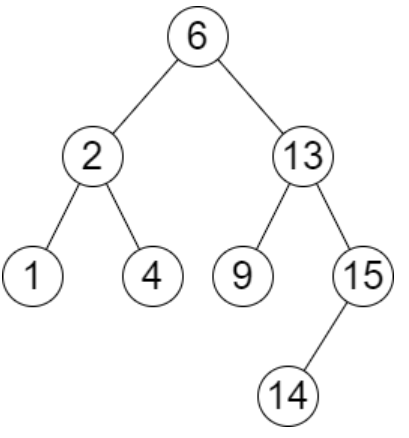
请你找出一个长度为n的二维答案数组answer，其中answer[i] = [mini, maxi]：

mini是树中小于等于queries[i]的最大值。如果不存在这样的值，则使用-1代替。

maxi是树中大于等于queries[i]的最小值。如果不存在这样的值，则使用 -1 代替。

返回数组 answer 。

示例 1 ：



输入：root = [6,2,13,1,4,9,15,null,null,null,null,null,null,14], queries = [2,5,16]

输出：[[2,2],[4,6],[15,-1]]

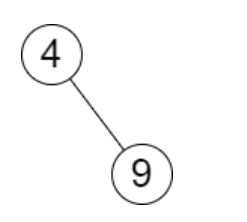
解释：按下面的描述找出并返回查询的答案：

- 树中小于等于 2 的最大值是 2 ，且大于等于 2 的最小值也是 2 。所以第一个查询的答案是 [2,2] 。

- 树中小于等于 5 的最大值是 4 ，且大于等于5的最小值是 6 。所以第二个查询的答案是 [4,6] 。

- 树中小于等于 16 的最大值是15，且大于等于16的最小值不存在。所以第三个查询的答案是 [15,-1] 。

示例 2 ：



输入：root = [4,null,9], queries = [3]

输出：[[-1,4]]

解释：树中不存在小于等于3的最大值，且大于等于3的最小值是4。所以查询的答案是 [-1,4] 。

提示：

树中节点的数目在范围[2, 105]内

1 <= Node.val <= 106

n == queries.length

1 <= n <= 105

1 <= queries[i] <= 106

# 分析