Problem 2476: Closest Nodes Queries in a Binary Search Tree

Problem Information

Difficulty: Medium
Acceptance Rate: 43.67%
Paid Only: No
Tags: Array, Binary Search, Tree, Depth-First Search, Binary Search Tree, Binary Tree

Problem Description

You are given the `root` of a **binary search tree** and an array `queries` of size `n` consisting of positive integers.

Find a **2D** array `answer` of size `n` where `answer[i] = [mini, maxi]`:

* `mini` is the **largest** value in the tree that is smaller than or equal to `queries[i]`. If a such value does not exist, add `-1` instead. * `maxi` is the **smallest** value in the tree that is greater than or equal to `queries[i]`. If a such value does not exist, add `-1` instead.

Return _the array_ `answer`.

Example 1:

![](https://assets.leetcode.com/uploads/2022/09/28/bstreeedrawioo.png)

Input: root = [6,2,13,1,4,9,15,null,null,null,null,null,null,14], queries = [2,5,16] **Output:** [[2,2],[4,6],[15,-1]] **Explanation:** We answer the queries in the following way: - The largest number that is smaller or equal than 2 in the tree is 2, and the smallest number that is greater or equal than 2 is still 2. So the answer for the first query is [2,2]. - The largest number that is smaller or equal than 5 in the tree is 4, and the smallest number that is greater or equal than 5 is 6. So the answer for the second query is [4,6]. - The largest number that is smaller or equal than 16 in the tree is 15, and the smallest number that is greater or equal than 16 does not exist. So the answer for the third query is [15,-1].

Example 2:

![](https://assets.leetcode.com/uploads/2022/09/28/bstttreee.png)

Input: root = [4,null,9], queries = [3] **Output:** [[-1,4]] **Explanation:** The largest number that is smaller or equal to 3 in the tree does not exist, and the smallest number that is greater or equal to 3 is 4. So the answer for the query is [-1,4].

Constraints:

* The number of nodes in the tree is in the range `[2, 105]`. * `1 <= Node.val <= 106` * `n == queries.length` * `1 <= n <= 105` * `1 <= queries[i] <= 106`

Code Snippets

C++:

```
/**
* Definition for a binary tree node.
* struct TreeNode {
* int val;
* TreeNode *left;
* TreeNode *right;
* TreeNode() : val(0), left(nullptr), right(nullptr) {}
* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
* TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
* };
*/
class Solution {
public:
vector<vector<int>> closestNodes(TreeNode* root, vector<int>& queries) {

}
};
```

Java:

```
/**
* Definition for a binary tree node.
* public class TreeNode {
* int val;
* TreeNode left;
* TreeNode right;
* TreeNode() {}
* TreeNode(int val) { this.val = val; }
* TreeNode(int val, TreeNode left, TreeNode right) {
* this.val = val;
* this.left = left;
* this.right = right;
* }
* }
*/
class Solution {
public List<List<Integer>> closestNodes(TreeNode root, List<Integer> queries)
{

}
}
```

Python3:

```
# Definition for a binary tree node.
# class TreeNode:
# def __init__(self, val=0, left=None, right=None):
# self.val = val
# self.left = left
# self.right = right
class Solution:
def closestNodes(self, root: Optional[TreeNode], queries: List[int]) ->
List[List[int]]:
```