

LeetCode

Explore

Problems

Interview

Contest

Discuss

Store

LeetCode is hiring! Apply NOW.

4

Description

Solution

Discuss (999+)

Submissions

99. Recover Binary Search Tree

Medium 5199 176 Add to List Share

You are given the `root` of a binary search tree (BST), where the values of **exactly** two nodes of the tree were swapped by mistake. *Recover the tree without changing its structure.*

Example 1:

1

3

2

→

3

1

2

Input: `root = [1,3,null,null,2]`
Output: `[3,1,null,null,2]`
Explanation: 3 cannot be a left child of 1 because 3 > 1. Swapping 1 and 3 makes the BST valid.

Example 2:

3

1

4

2

→

2

1

4

3

Input: `root = [3,1,4,null,null,2]`
Output: `[2,1,4,null,null,3]`
Explanation: 2 cannot be in the right subtree of 3 because 2 < 3. Swapping 2 and 3 makes the BST valid.

Constraints:

- The number of nodes in the tree is in the range `[2, 1000]`.
- $-2^{31} \leq \text{Node.val} \leq 2^{31} - 1$

Follow up: A solution using $O(n)$ space is pretty straight-forward. Could you devise a constant $O(1)$ space solution?

Accepted 320,708 Submissions 659,881

Seen this question in a real interview before?

Yes

No

Companies

i

Related Topics

Java

Autocomplete

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

```
/**
 * 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 void recoverTree(TreeNode root) {
    }
}
```

Problems

Pick One

Prev

99/2247

Next

Console

Contribute i

Run Code

Submit

https://leetcode.com/problems/recover-binary-search-tree/

1/1