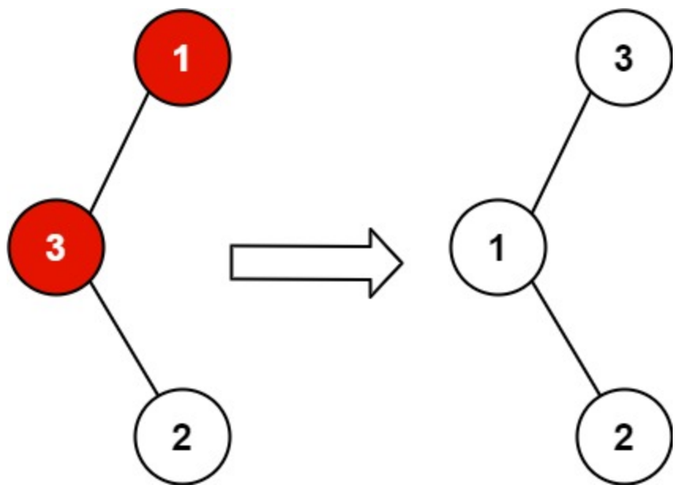
99. Recover Binary Search Tree

Medium 4851 167 Add to List Share

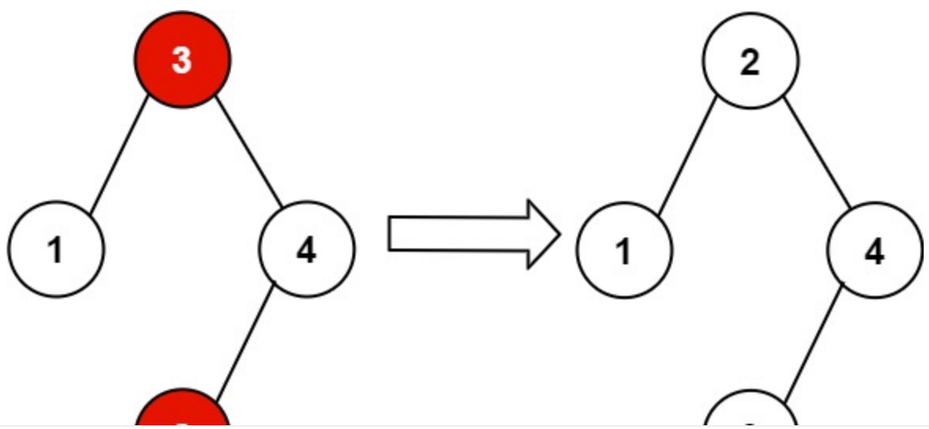
You are given the `root` of a binary search tree (BST), where the values of **exactly** two nodes of the tree were swapped by mistake.
Recover the tree without changing its structure.

Example 1:



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:



Java Autocomplete

```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 {

    private TreeNode prev = null;
    private TreeNode one = null;
    private TreeNode two = null;

    public void recoverTree(TreeNode root) {
        inorder(root);
        int temp = one.val;
        one.val = two.val;
        two.val = temp;
    }

    private void inorder(TreeNode curr){

        if(curr == null)
            return;

        inorder(curr.left);

        //Do the business logics here
        if(prev != null && prev.val > curr.val){
            if(one == null)
                one = prev;
            two = curr;
        }

        prev = curr;

        inorder(curr.right);
    }
}
```

Your previous code was restored from your local storage. Reset to default