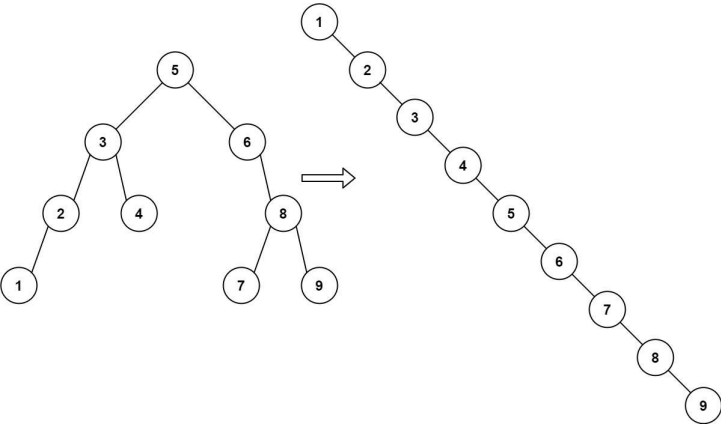


897. Increasing Order Search Tree

Easy 3136 629 Add to List Share

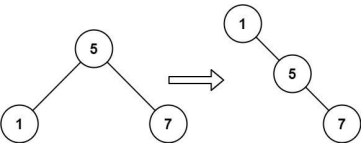
Given the root of a binary search tree, rearrange the tree in **in-order** so that the leftmost node in the tree is now the root of the tree, and every node has no left child and only one right child.

Example 1:



Input: root = [5,3,6,2,4,null,8,1,null,null,null,7,9]
Output: [1,null,2,null,3,null,4,null,5,null,6,null,7,null,8,null,9]

Example 2:



Input: root = [5,1,7]
Output: [1,null,5,null,7]

Constraints:

- The number of nodes in the given tree will be in the range [1, 100] .
- 0 <= Node.val <= 1000

Accepted 208,404 Submissions 266,337

Seen this question in a real interview before?

Yes

No

Companies i

Related Topics

```
1 /**
2  * Definition for a binary tree node.
3  * public class TreeNode {
4  *     int val;
5  *     TreeNode left;
6  *     TreeNode right;
7  *     TreeNode() {}
8  *     TreeNode(int val) { this.val = val; }
9  *     TreeNode(int val, TreeNode left, TreeNode right) {
10 *         this.val = val;
11 *         this.left = left;
12 *         this.right = right;
13 *     }
14 * }
15 */
16 class Solution {
17     public TreeNode increasingBST(TreeNode root) {
18     }
19 }
20
```