

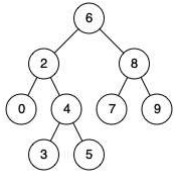
235. Lowest Common Ancestor of a Binary Search Tree

Easy 🍌 4981 🍌 169 ❤️ Add to List 📄 Share

Given a binary search tree (BST), find the lowest common ancestor (LCA) of two given nodes in the BST.

According to the definition of LCA on Wikipedia: "The lowest common ancestor is defined between two nodes p and q as the lowest node in T that has both p and q as descendants (where we allow **a node to be a descendant of itself**)."

Example 1:

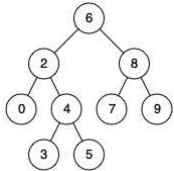


Input: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 8

Output: 6

Explanation: The LCA of nodes 2 and 8 is 6.

Example 2:



Input: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 4

Output: 2

Explanation: The LCA of nodes 2 and 4 is 2, since a node can be a descendant of itself according to the LCA definition.

Example 3:

Input: root = [2,1], p = 2, q = 1

Output: 2

Constraints:

- The number of nodes in the tree is in the range $[2, 10^5]$.
- $-10^9 \leq \text{Node.val} \leq 10^9$
- All Node.val are **unique**.
- $p \neq q$
- p and q will exist in the BST.

Accepted 682,065 Submissions 1,214,872

Seen this question in a real interview before?

Yes

No

Companies 🍌 i

Related Topics

Similar Questions

```
1  /**
2   * Definition for a binary tree node.
3   * public class TreeNode {
4   *     int val;
5   *     TreeNode left;
6   *     TreeNode right;
7   *     TreeNode(int x) { val = x; }
8   * }
9   */
10
11 class Solution {
12     public TreeNode lowestCommonAncestor(TreeNode root, TreeNode p, TreeNode q) {
13
14     }
15 }
```