

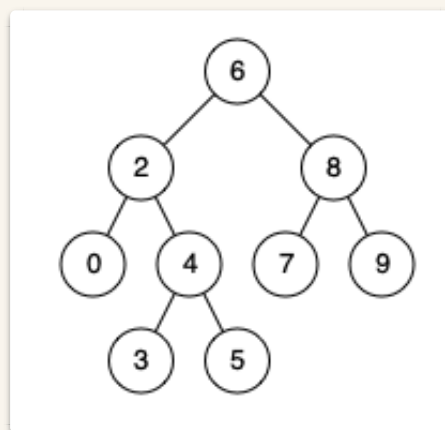
235 BST的最近公共祖先

题目描述

给定一个二叉搜索树，找到该树中两个指定节点的最近公共祖先。

百度百科中最近公共祖先的定义为：“对于有根树 T 的两个结点 p、q，最近公共祖先表示为一个结点 x，满足 x 是 p、q 的祖先且 x 的深度尽可能大（一个节点也可以是它自己的祖先）。”

例如，给定如下二叉搜索树: root = [6,2,8,0,4,7,9,null,null,3,5]



示例 1:

- 1 输入: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 8
- 2 输出: 6
- 3 解释: 节点 2 和节点 8 的最近公共祖先是 6。

示例 2:

- 1 输入: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 4
- 2 输出: 2
- 3 解释: 节点 2 和节点 4 的最近公共祖先是 2，因为根据定义最近公共祖先节点可以为节点本身。

说明:

- 所有节点的值都是唯一的。
- p、q 为不同节点且均存在于给定的二叉搜索树中。

代码

```
1 # Definition for a binary tree node.
2 # class TreeNode:
3 #     def __init__(self, x):
4 #         self.val = x
5 #         self.left = None
6 #         self.right = None
7
8 class Solution:
9     def lowestCommonAncestor(self, root: 'TreeNode', p: 'TreeNode', q:
    'TreeNode') -> 'TreeNode':
10         if not root:
11             return root
12         if p.val > q.val:
13             return self.lowestCommonAncestor(root, q, p)
14         if root.val > q.val:
15             return self.lowestCommonAncestor(root.left, p, q)
16         elif root.val < p.val:
17             return self.lowestCommonAncestor(root.right, p, q)
18         elif root.val >= p.val and root.val <= q.val:
19             return root
```

成功 [显示详情 >](#)

执行用时：**116 ms**, 在Lowest Common Ancestor of a Binary Search Tree的Python3提交中击败了**72.31%**的用户

内存消耗：**17.3 MB**, 在Lowest Common Ancestor of a Binary Search Tree的Python3提交中击败了**92.24%**的用户

进行下一个挑战：

[二叉树的右视图](#)

[二叉树的序列化与反序列化](#)

[拆分二叉搜索树](#)

炫耀一下：[!\[\]\(e474458956c9a37fbf9586ddb60a7fa1_img.jpg\)](#) [!\[\]\(4d1d3f2547aeece54bb6babd23f4121b_img.jpg\)](#) [!\[\]\(ec45aa71601db5755c5e2662ad427708_img.jpg\)](#) [!\[\]\(8f6ad92394b094baf6a51f98af6c5abc_img.jpg\)](#) [!\[\]\(f9f9ea03f3583772c68aa68b93940c29_img.jpg\)](#)

提交时间	状态	执行用时	内存消耗	语言
几秒前	通过	116 ms	17.3 MB	python3
几秒前	通过	128 ms	17.2 MB	python3