1650. Lowest Common Ancestor of a Binary Tree III

Medium 516 15 Add to List Share

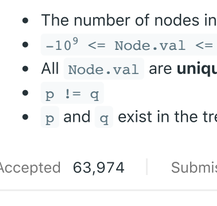
Given two nodes of a binary tree `p` and `q`, return *their lowest common ancestor (LCA)*.

Each node will have a reference to its parent node. The definition for `Node` is below:

```
class Node {
    public int val;
    public Node left;
    public Node right;
    public Node parent;
}
```

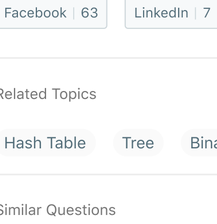
According to the **definition of LCA on Wikipedia**: "The lowest common ancestor of two nodes p and q in a tree T is the lowest node that has both p and q as descendants (where we allow **a node to be a descendant of itself**)."

Example 1:




```
Input: root = [3,5,1,6,2,0,8,null,null,7,4], p = 5, q = 1
Output: 3
Explanation: The LCA of nodes 5 and 1 is 3.
```

Example 2:




```
Input: root = [3,5,1,6,2,0,8,null,null,7,4], p = 5, q = 4
Output: 5
Explanation: The LCA of nodes 5 and 4 is 5 since a node can be a descendant of itself according to the LCA definition.
```

Example 3:

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

Constraints:

- The number of nodes in the tree is in the range `[2, 10^5]`.
- `-10^9 <= Node.val <= 10^9`
- All `Node.val` are **unique**.
- `p != q`
- `p` and `q` exist in the tree.

Accepted 63,974 | Submissions 82,654

Seen this question in a real interview before? Yes No

Companies

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Facebook | 63 LinkedIn | 7 Microsoft | 4 Amazon | 4 Google | 2

Related Topics

Hash Table Tree Binary Tree

Similar Questions

Lowest Common Ancestor of a Binary Search Tree	Easy
Lowest Common Ancestor of a Binary Tree	Medium
Lowest Common Ancestor of a Binary Tree II	Medium
Lowest Common Ancestor of a Binary Tree IV	Medium

Hide Hint 1

Store the path from p to the root.

Hide Hint 2

Traverse the path from q to the root, the first common point of the two paths is the LCA.

Java Autocomplete

```java
/*
// Definition for a Node.
class Node {
    public int val;
    public Node left;
    public Node right;
    public Node parent;
};
*/

class Solution {
    public Node lowestCommonAncestor(Node p, Node q) {

    }
}
```