```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode(int x) { val = x; }
 * }
 */
class Solution {
    public TreeNode lowestCommonAncestor(TreeNode root, TreeNode p, TreeNode q) {
        if (root == null || root == p || root == q) {
            return root;
        }

        TreeNode n1 = lowestCommonAncestor(root.left, p, q);
        TreeNode n2 = lowestCommonAncestor(root.right, p, q);
        if (n1 != null && n2 != null) {
            return root;
        } else if (n1 != null) {
            return n1;
        } else if (n2 != null) {
            return n2;
        }
        return null;
    }
}
```

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode(int x) { val = x; }
 * }
 */
class Solution {
    public TreeNode lowestCommonAncestor(TreeNode root, TreeNode p, TreeNode q) {
        if (root == null)
            return null;
        if (root.val > p.val && root.val > q.val) {
            return lowestCommonAncestor(root.left, p, q);
        } else if (root.val < p.val && root.val < q.val) {
            return lowestCommonAncestor(root.right, p, q);
        }
        return root;
    }
}
```