```swift
import Foundation
/// **剑指 Offer 68 - I. 二叉搜索树的最近公共祖先**
/// 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）。”
///
/// 示例 1:
/// 输入: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 8
/// 输出: 6
/// 解释: 节点 2 和节点 8 的最近公共祖先是 6。
public class TreeNode {
    public var val: Int
    public var left: TreeNode?
    public var right: TreeNode?

    public init() { self.val = 0; self.left = nil; self.right = nil; }
    public init(_ val: Int) { self.val = val; self.left = nil; self.right =
     nil; }
    public init(_ val: Int, _ left: TreeNode?, _ right: TreeNode?) {
        self.val = val
        self.left = left
        self.right = right
    }
}

class Solution {
    public func lowestCommonAncestor(_ root: TreeNode?,
                                     _ p: TreeNode?,
                                     _ q: TreeNode?) -> TreeNode? {
        // 若根节点为空，则直接返回 nil
        if root == nil {
            return nil
        }
        // 若根节点值大于 p 和 q 的值，则最近公共祖先在左子树中
        if root!.val > p!.val && root!.val > q!.val {
            return lowestCommonAncestor(root?.left, p, q)
        }
        // 若根节点值小于 p 和 q 的值，则最近公共祖先在右子树中
        if root!.val < p!.val && root!.val < q!.val {
            return lowestCommonAncestor(root?.right, p, q)
        }
        // 根节点值在 p 和 q 的值之间，或者其中一个节点为根节点，则最近公共祖先为根节点
        return root
    }
}
```