235. Lowest Common Ancestor of a Binary Search Tree

作者：qianrong wu

思路

首选root，p，q不能为空；然后对于树中从root开始的节点：
如果p和q的值如果都小于root的值，那么它们的最低公共祖先一定在root的左子树；如果p和q的值如果都大于root的值，那么它们的最低公共祖先一定在root的右子树；其他情况则说明最低公共祖先就是root节点。如此从root往下循环判断。

Python Code

```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, x):
#         self.val = x
#         self.left = None
#         self.right = None

class Solution:
    def lowestCommonAncestor(self, root, p, q):
        """
        :type root: TreeNode
        :type p: TreeNode
        :type q: TreeNode
        :rtype: TreeNode
        """
        if not root or not p or not q:
            return none
        if max(p.val, q.val) < root.val:
            return self.lowestCommonAncestor(root.left, p, q)
        elif min(p.val, q.val) > root.val:
            return self.lowestCommonAncestor(root.right, p, q)
        else:
            return root
```

总结

1. 从上往下找
2. 按照大小比较分类讨论
3. 循环