1. 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）。”

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:

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。

示例 2:

输入: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 4

输出: 2

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。

解决方案：

根据二叉搜索树的性质，我们可以得知，左子树所有节点val值都小于根节点val值，右子树所有节点val值都大于根节点val值。

我们可以依次比较根节点的val值和p、q节点的val值

1. 当p <= root <= q或者 q <= root <= p时，我们可以得出p、q的公共祖先为root节点
2. 当p、q节点都小于root节点，那么可以得出p、q的公共祖先在左子树中，那么可以递归地调用该方法，将根节点替换为左子节点
3. 当p、q节点都大于root节点，那么可以得出p、q的公共祖先在右子树中，那么可以递归地调用该方法，将根节点替换为右子节点