

# 二叉树的祖先问题

## 二叉树的最近公共祖先

**公共祖先的定义：**对于有根树  $T$  的两个节点  $p$ 、 $q$ ，最近公共祖先表示为一个节点  $x$ ，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）

**方法：**由于要使公共节点  $x$  深度尽可能大，结合递归类似栈的工作方式，此题想到用递归法

**递归三部曲：**

(1) **递归参数和返回类型：**参数是根节点和  $p$ 、 $q$  两个节点，返回类型也是节点，因此直接用主函数做递归；

(2) **递归终止条件：**当递归到  $root$  为空（没找到）或  $root$  为  $p$  或  $root$  为  $q$  时（不能再递归更深了，更深递归得到的结果必不可能是公共祖先），返回  $root$ （对应了以上三种情况）；

(3) **递归逻辑：**对根节点的左/右节点分别递归

**递归结果的处理（递归函数的变量分别为  $left$ 、 $right$ ）：**

- (1) 如果  $left$ 、 $right$  均为空，说明没找到  $p$  和  $q$ ，返回空；
- (2) 如果  $left$  为空，当不属于情况 (1) 时，说明在  $right$  中找到了答案，返回  $right$ ；
- (3) 如果  $right$  为空，当不属于情况 (1) 时，说明在  $left$  中找到了答案，返回  $left$ ；
- (4) 不属于情况 (1) - (3) 时，说明在  $left$  和  $right$  中都找到了答案，说明  $p$  和  $q$  分布在根节点的两侧，此时根节点是它们的公共祖先，返回  $root$

题目：

### 236. 二叉树的最近公共祖先（中等）

## 二叉搜索树的最近公共祖先

可以用“二叉树的最近公共祖先”的方法，这属于查找二叉树最近公共祖先的通解

**本题方法：**充分利用二叉搜索树的性质，分为以下三种情况：

- (1) 如果要查找的两个节点值都比  $root$  节点小，说明最近公共祖先出现在  $root$  的左子树中， $root$  往左；
- (2) 如果都比  $root$  节点大，说明最近公共祖先出现在  $root$  的右子树中， $root$  往右；

(3) 其余情况，要么root就是两个节点中的其中一个，要么一个比root小，一个比root大，统一作为else情况，此时root就是它们的公共祖先

**循环设置形式：**当找到某节点root是公共祖先时，即break退出循环，返回root；否则就一直进行循环判断 (**while(true)**)

题目：

### 235. 二叉搜索树的最近公共祖先（中等）