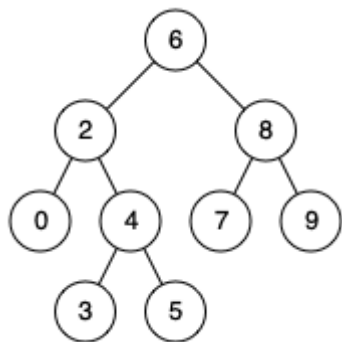


第16题——二叉搜索树的最近共祖先

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。

理解题意

首先，作为一个二叉搜索树，它本身是有序的，并且不存在重复值

所以，对于一个结点，有以下三种结果：

1. 该节点的值既大于p也大于q，则最近共祖先一定在它的左子树中，我们返回左子树。
2. 该节点的值既小于p也小于q，则最近共祖先一定在它的右子树中，我们返回右子树
3. 该节点的值恰好处于p q之间，则根据二叉搜索树的结构，该节点就是p和q的最近共祖先。退出循环返回该节点即可。

代码如下：

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */

struct TreeNode* lowestCommonAncestor(struct TreeNode* root, struct TreeNode* p, struct TreeNode* q) {
    struct TreeNode *res=root;
    while(1)
    {
        //两个值小于res，则一定在左子树
        if(p->val<res->val && q->val<res->val){
            res=res->left;
        }
        //两个值都大于res，则一定在右子树
        else if(p->val>res->val && q->val>res->val){
            res=res->right;
        }
        //都不是，那么该值在两个结点之间，找到共祖先
        else{
            break;
        }
    }
    return res;
}
```

作者：sakura7301

链接：<https://leetcode-cn.com/problems/er-cha-sou-suo-shu-de-zui-jin-gong-gong-zu-xian-lcof/solution/yi-ge-whilejie-jue-by-sakura7301-9qdx/>

来源：力扣（LeetCode）

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。