```
/*
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。

来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/lowest-common-ancestor-of-a-binary-search-tree
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
 */
```

分析:

- 若p, q的val都小于当前的节点的val,则最近公共祖先一定出现在左子树;
- 若p, q的val都大于当前的节点的val,则最近公共祖先一定出现在右子树;
- 其他情况下,当前节点就是最近公共祖先.

方法一:C++_左右根遍历

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
```

```
 * };
 */
class Solution
{

    private:

        void helper (   TreeNode*&  ret_val     ,
                        TreeNode*   node        ,
                        TreeNode*   p           ,
                        TreeNode*   q
        )
        {
            if(node==NULL)
            {
                return;
            }

            if(node->val < p->val && node->val < q->val)
            {
                helper(ret_val,node->right,p,q);
                return;
            }

            if(node->val > p->val && node->val > q->val)
            {
                helper(ret_val,node->left,p,q);
                return;
            }
            ret_val = node;
            return;




        }


    public:
        TreeNode* lowestCommonAncestor( TreeNode*   root    ,
                                        TreeNode*   p       ,
                                        TreeNode*   q
                                       )
        {
            TreeNode* ret_val = NULL;
            helper(ret_val,root,p,q);
            return ret_val;

        }
};


/*
执行结果:
通过
显示详情
执行用时 :40 ms, 在所有 cpp 提交中击败了82.59% 的用户
```

```
内存消耗 :25.5 MB, 在所有 cpp 提交中击败了96.73%的用户
*/
```

AlimyBreak
2019.12.07