**题目要求：Given a binary search tree (BST), find the lowest common ancestor (LCA) of two given nodes in the BST.**

According to the [definition of LCA on Wikipedia](https://en.wikipedia.org/wiki/Lowest_common_ancestor): “The lowest common ancestor is defined between two nodes v and was the lowest node in T that has both v and was descendants (where we allow a node to be a descendant of itself).”

\_\_\_\_\_\_\_6\_\_\_\_\_\_

/ \

\_\_\_2\_\_ \_\_\_8\_\_

/ \ / \

0 \_4 7 9

/ \

3 5

For example, the lowest common ancestor (LCA) of nodes 2 and 8 is 6. Another example is LCA of nodes 2 and 4 is 2, since a node can be a descendant of itself according to the LCA definition.

My solution(Your runtime beats 36.87% of cppsubmissions,40ms):

struct TreeNode {

int val;

TreeNode \*left;

TreeNode \*right;

TreeNode(int x) : val(x), left(NULL), right(NULL) {}

};

class Solution {

public:

TreeNode\* lowestCommonAncestor(TreeNode\* root, TreeNode\* p, TreeNode\* q) {

if (root == NULL)return NULL;

**if (p->val<=root->val&&root->val<= q->val||p->val>= root->val&&root->val >= q->val)return root;**

if (p->val > root->val&&root->val < q->val){

return lowestCommonAncestor(root->right, p, q);

}

if (p->val < root->val&&root->val > q->val){

return lowestCommonAncestor(root->left, p, q);

}

}

};

分析:

利用二叉搜索树的性质，如果p,q都比根结点大，说明p,q两节点的最小父结点在根结点的右边，否则在左边,当满足条件时即得到符合条件的结点.