# 68. 树中两个节点的最低公共祖先

## 68.1 二叉查找树

### 题目链接

[Leetcode : 235. Lowest Common Ancestor of a Binary Search Tree](https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-search-tree/description/)

### 解题思路

在二叉查找树中，两个节点 p, q 的公共祖先 root 满足 root.val >= p.val && root.val <= q.val。

public TreeNode lowestCommonAncestor(TreeNode root, TreeNode p, TreeNode q) {
 if (root == null)
 return root;
 if (root.val > p.val && root.val > q.val)
 return lowestCommonAncestor(root.left, p, q);
 if (root.val < p.val && root.val < q.val)
 return lowestCommonAncestor(root.right, p, q);
 return root;
}

## 68.2 普通二叉树

### 题目链接

[Leetcode : 236. Lowest Common Ancestor of a Binary Tree](https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-tree/description/)

### 解题思路

在左右子树中查找是否存在 p 或者 q，如果 p 和 q 分别在两个子树中，那么就说明根节点就是最低公共祖先。

public TreeNode lowestCommonAncestor(TreeNode root, TreeNode p, TreeNode q) {
 if (root == null || root == p || root == q)
 return root;
 TreeNode left = lowestCommonAncestor(root.left, p, q);
 TreeNode right = lowestCommonAncestor(root.right, p, q);
 return left == null ? right : right == null ? left : root;
}