LeetCode\_315\_CountOfSmallerNumbersAfterSelf—Hard

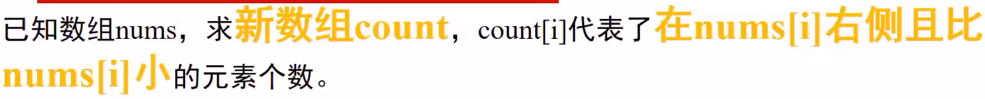
# 题目描述

LeetCode\_315\_CountOfSmallerNumbersAfterSelf—Hard

难度：**Hard**

<https://leetcode.com/problems/count-of-smaller-numbers-after-self/description/>

You are given an integer array nums and you have to return a new counts array. The counts array has the property where counts[i] is the number of smaller elements to the right of **nums[i]**.



Example:

Input: [5,2,6,1]

Output: [2,1,1,0]

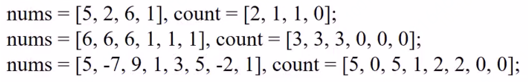
Explanation:

To the right of 5 there are 2 smaller elements (2 and 1).

To the right of 2 there is only 1 smaller element (1).

To the right of 6 there is 1 smaller element (1).

To the right of 1 there is 0 smaller element.

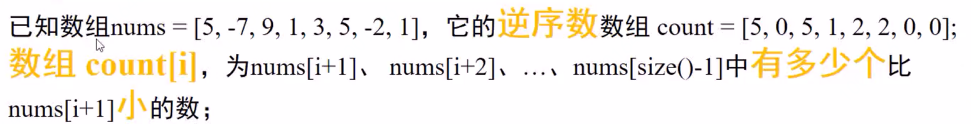


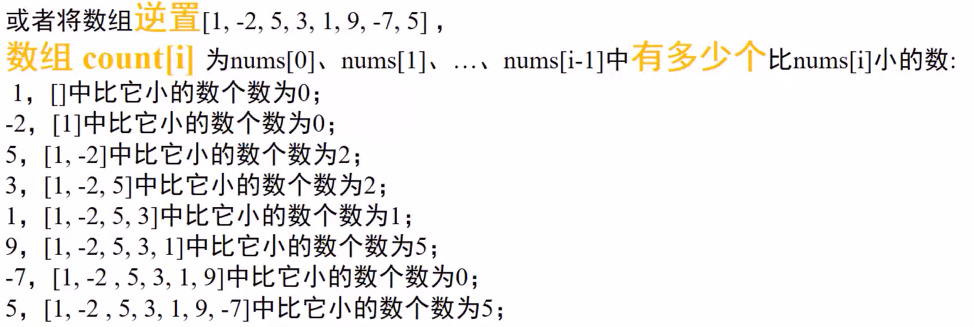
# 解决思路

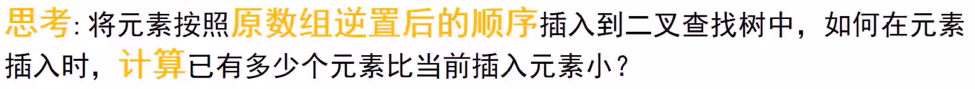
解决方法1：**归并**；(遇到时再说)

解决方法2：**二叉查找树**；(又称二叉排序树)重点介绍该方法。

## 分析思路：

**需要逆置**：





TreeNodeWithCount : count用来记录左子树的节点数量。

/\*\*

\* 定义一个带有count属性的二叉树结构

\*/

class TreeNodeWithCount{

int value;

int count;/\*用于记录当前节点的左子树的节点数目

(包括叶子节点和非叶子节点)；意义：当前比该节点值小的节点数目\*/

TreeNodeWithCount left;

TreeNodeWithCount right;

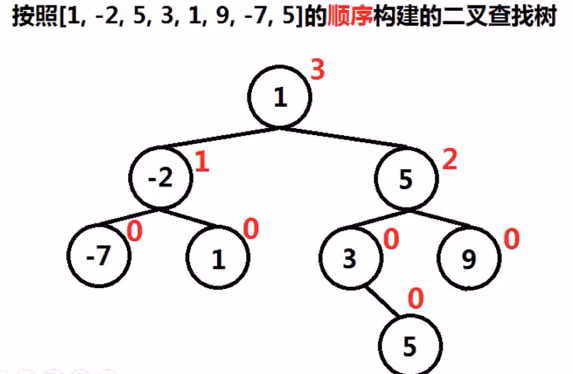
public TreeNodeWithCount(int value){

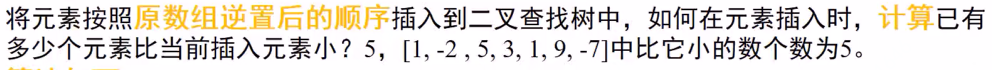
this.value = value;

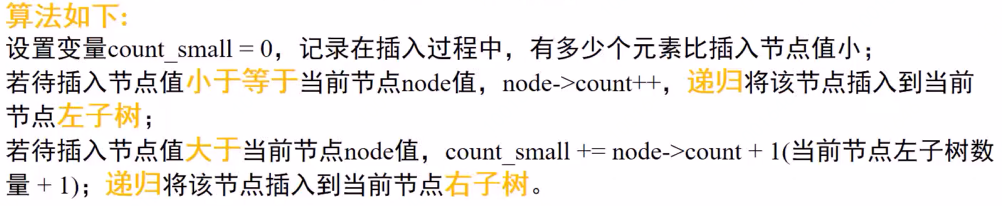
}

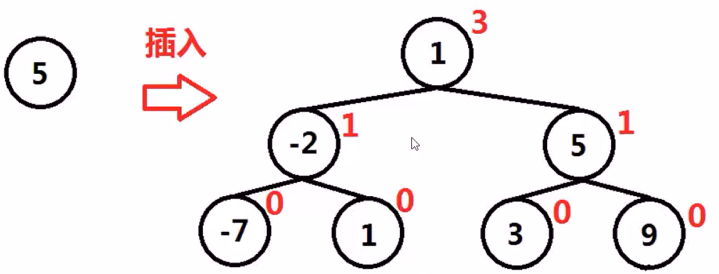
}











示例：

