Leetcode (python) 二叉搜索树 困难（只做了315）

315[计算右侧小于当前元素的个数](https://leetcode-cn.com/problems/count-of-smaller-numbers-after-self/)

题目：给定一个整数数组 *nums*，按要求返回一个新数组 *counts*。数组 *counts* 有该性质： counts[i] 的值是  nums[i] 右侧小于 nums[i] 的元素的数量。

**示例:**

**输入:** [5,2,6,1]

**输出:** [2,1,1,0]

**解释:**

5 的右侧有 **2** 个更小的元素 (2 和 1).

2 的右侧仅有 **1** 个更小的元素 (1).

6 的右侧有 **1** 个更小的元素 (1).

1 的右侧有 **0** 个更小的元素.

思路：利用 二分查找来做！ 对列表元素倒着挨个插入一个列表构成有序列表，插入时该节点的插入位置即为比该节点小的节点数目 要注意的是，要用bisect\_left来找，bisect\_left和bisect\_right的不同在于对同一数值的元素 bisect\_left返回左值，bisect\_right返回右值 如 r = [1,2,3,4,5],插入4， bisect\_left 返回3，bisect\_right返回4 """

代码：

1. **class** Solution(object):
2. **def** countSmaller(self, nums):
3. """
4. :type nums: List[int]
5. :rtype: List[int]
6. """
7. **import** bisect
8. r = []
9. res = [0] \* len(nums)
10. **for** i **in** range(len(nums) - 1, -1, -1):
11. bisect.insort(r, nums[i])
12. res[i] = bisect.bisect\_left(r, nums[i])
13. **return** res