# 题目

给你一个数组nums，对于其中每个元素nums[i]，请你统计数组中比它小的所有数字的数目。

换而言之，对于每个nums[i]你必须计算出有效的j的数量，其中j满足j != i且nums[j] < nums[i]。

以数组形式返回答案。

示例 1：

输入：nums = [8,1,2,2,3]

输出：[4,0,1,1,3]

解释：

对于 nums[0]=8 存在四个比它小的数字：（1，2，2 和 3）。

对于 nums[1]=1 不存在比它小的数字。

对于 nums[2]=2 存在一个比它小的数字：（1）。

对于 nums[3]=2 存在一个比它小的数字：（1）。

对于 nums[4]=3 存在三个比它小的数字：（1，2 和 2）。

示例 2：

输入：nums = [6,5,4,8]

输出：[2,1,0,3]

示例 3：

输入：nums = [7,7,7,7]

输出：[0,0,0,0]

提示：

2 <= nums.length <= 500

0 <= nums[i] <= 100

# 分析

## 方法一：暴力法

class Solution {

public:

vector<int> smallerNumbersThanCurrent(vector<int>& nums) {

vector<int> counts(nums.size(), 0);

for (int i = 0; i < nums.size(); ++i) {

for (int j = 0; j < nums.size(); ++j) {

if (nums[j] < nums[i]) {

++counts[i];

}

}

}

return counts;

}

};

## 方法二：排序法

class Solution {

public:

vector<int> smallerNumbersThanCurrent(vector<int>& nums) {

vector<int> sorted = nums;

sort(sorted.begin(), sorted.end());

vector<int> result(nums.size());

for (int i = 0; i < nums.size(); ++i) {

int count = 0;

for (int j = 0; j < sorted.size(); ++j) {

if (sorted[j] < nums[i]) {

count++;

} else {

break;

}

}

result[i] = count;

}

return result;

}

};