[每日 LeetCode] 665. Non-decreasing Array

作者：Hanseltu
原文链接：https://ld246.com/article/1551795196326
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

Description:

Given an array with n integers, your task is to check if it could become non-decreasing by mo ifying **at most** 1 element.

We define an array is non-decreasing if array[i] <= array[i + 1] holds for every i (1 <= i < n).

Example 1:

Input: [4,2,3]
Output: True
Explanation: You could modify the first 4 to 1 to get a non-decreasing array.

Example 2:

Input: [4,2,1]
Output: False
Explanation: You can't get a non-decreasing array by modify at most one element.

Note:

The n belongs to [1, 10,000].

思路：本题要求判断能否最多只修改数组中的一个数，使得数组成为非递减数组。乍一看没什么规律目前的想法只有根据不同的测试用例分类来判断，即对可以最多一步修改到位的测试用例进行归纳总。总共有以下几种情况是可能修改一次就到位的（也是根据修改元素的位置）：

(首先先统计元素递减的对数（从下标1开始），如[1,4,2,5,1],有两组元素递减，只有一组元素递减的组才可能满足题意)

1 修改的元素在最前面，如[4,1,2,3]，修改4即可；

2 修改的元素在中间，有两种情况(注意位置)

（1）修改元素的前后是非递减的，如[1,2,4,3,5]，修改4即可；

（2）修改元素的前后是非递减的，如[-1,4,2,3]，修改4即可；

3 修改的元素在最后，如[1,2,3,4,1]，修改最后的1即可。

C++代码

```
class Solution {
public:
    bool checkPossibility(vector<int>& nums) {
        int modify = 0, index = 0, i;
        for(int i=1; i<nums.size(); i++)
        {
            if(nums[i] < nums[i-1])
            {
                index = i;
                modify++;
            }
        }
```

```
        if(modify == 0){
            return true;
        }
        if(modify > 1){
             return false;
        }
        if(index==1||index==nums.size()-1||nums[index+1]>=nums[index-1]||nums[index]>=nu
s[index-2]){
            return true;
        }
        return false;
    }
};
```

运行时间：36ms

运行内存：12.2M