

# 283. 移动零

地址：[🔗 移动零](#)

题目：

- English：

283. Move Zeroes

难度 简单 1109

Given an integer array `nums`, move all `0`'s to the end of it while maintaining the relative order of the non-zero elements.

**Note** that you must do this in-place without making a copy of the array.

Example 1:

**Input:** `nums = [0,1,0,3,12]`

**Output:** `[1,3,12,0,0]`

Example 2:

**Input:** `nums = [0]`

**Output:** `[0]`

Constraints:

- `1 <= nums.length <= 104`
- `-231 <= nums[i] <= 231 - 1`

- 中文：

283. 移动零

难度 简单 1109

给定一个数组 `nums`，编写一个函数将所有 `0` 移动到数组的末尾，同时保持非零元素的相对顺序。

示例:

输入: `[0,1,0,3,12]`

输出: `[1,3,12,0,0]`

说明:

- 必须在原数组上操作，不能拷贝额外的数组。
- 尽量减少操作次数。

## 思路 1: 利用 `inserting_index` 在原数组上操作

### 分析

- ★ 用变量 `zero_index` 记录非零数将要插入要代替 `0` 元素的位置
- ★ 遍历数组,将非零值赋在要代替 `0` 元素的位置
- ★ 对原非零数位置 赋 `0`

### Code

```
1 # Python3
2 # Did Time : 2021 - 07 - 07
3
```

```

4
5 class Solution:
6
7
8     def moveZeroes(self, nums: List[int]) -> None:
9         """
10         Do not return anything, modify nums in-place instead.
11         """
12         zero_index = 0    # position to insert the non zero numbe
13
14
15         for i in range(0, len(nums)):    # iterate through the nums array
16             if(nums[i] != 0):            # if ith value not equals 0, set its value to zero position
17                 nums[zero_index] = nums[i]
18                 if(i != zero_index):    # when ith is not the zero index, set ith value as 0
19                     nums[i] = 0
20                 zero_index += 1          # move to next zero index

```

```

1 // Java
2 // Did Time : 2021 - 07 - 07
3 class Solution {
4
5
6     public void moveZeroes(int[] nums) {
7         int j = 0;
8         for (int i = 0; i < nums.length; ++i) {
9             if (nums[i] != 0) {
10                 nums[j] = nums[i];
11                 if (i != j) {
12                     nums[i] = 0;
13                 }
14                 j++;
15             }
16         }
17     }
18 }

```

## 复杂度分析：

1. 时间复杂度 :  $O(n)$
2. 空间复杂度 :  $O(1)$