



链滴

# 移动零算法

作者: [selfjt](#)

原文链接: <https://ld246.com/article/1573042387767>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



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。

## 场景

### 中文描述

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。

### English

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

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。

## 代码示例

### 方法一

```
func moveZeroes(nums []int) []int {  
    count := 0
```

```
for i := 0; i < len(nums); i++ {
    if nums[i] == 0 {
        count++
    } else {
        if count > 0 {
            nums[i-count], nums[i] = nums[i], nums[i-count]
        }
    }
}
return nums
}
```

执行用时：224 ms, 内存消耗：15.2 MB

## 方法二

```
func moveZeroes2(nums []int) []int {
    count := 0
    for i := 0; i < len(nums); i++ {
        //找出不为0的个数
        if nums[i] != 0 {
            nums[count] = nums[i]
            count++
        }
    }
    //把不为0后面重置为0
    for i := count; i < len(nums); i++ {
        nums[i] = 0
    }
    return nums
}
```

执行用时：132 ms, 内存消耗：14.8 MB

## 记录自己学习算法.

- 大家有什么更好的算法可以留言.

## 源码

- [点击查看源码](#)