

# 移动零

## 方法一：双指针

使用**双指针**，left与right，如果右指针指向的数不为0，则交换左右指针的数字，此时右指针和左指针都向右移动，如果右指针指向的数为0，则左指针不动，右指针向右移动。

当左指针指向的是第一个0的位置，右指针指向的是第一个非0的位置时，进行交换

1. 左指针左边均为非零数；
2. 右指针左边直到左指针处均为零。

```
class Solution {
    public void moveZeroes(int[] nums) {
        int n = nums.length, left = 0, right = 0;
        while(right < n){
            if(nums[right] != 0){
                swap(nums, left, right);
                left++;
            }
            right++;
        }
    }

    public void swap(int[] nums, int left, int right){
        int temp = nums[right];
        nums[right] = nums[left];
        nums[left] = temp;
    }
}
```

## 方法二：覆盖

i统计数组中0出现的个数，当数组中出现不为0的数时，把当前数字放到相应的0的位置，并把当前位置归0。

注意：i != 0的限制条件，没有此限制条件时，当数组中全部不为0时，输出结果中的数组会变为0

```
class Solution{
    public void moveZeroes(int[] nums){
        int i = 0;
        for (int j = 0; j < nums.length; j++){
            if(nums[j] == 0){
                i++;
            }else if(i != 0){
                nums[j-i] = nums[j];
                nums[j] = 0;
            }
        }
    }
}
```