

283 移动零

Label: 双指针

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。

保证相对顺序的话，就不可能直接先进行排序

- 双指针 两次遍历

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

        for(int p = i; p < nums.length; p++){ // 替换尾部非0元素
            nums[p] = 0;
        }
    }
}
```

- 双指针 一次遍历

```
class Solution { // 类似于快排，把不等于0的放在左边，把等于0的放在右边
    public void moveZeroes(int[] nums) {

        int i = 0 ;
        for(int j = 0; j < nums.length ; j++){
            if (nums[j] != 0) {
                int temp = nums[i];
                nums[i] = nums[j];
                nums[j] = temp; //本来这里可以直接使用0，但是为了防止只有一个非0元素的情况，所以必须使用temp
                i++;
            }
        }
    }
}
```