[LeetCode] Move Zeroes - 整数数组处理问题

目录：

一.Move Zeroes

题目概述：

Given an arraynums, write a function to move all 0's to the end of it while maintaining the relative order of the non-zero elements.For example, givennums = [0, 1, 0, 3, 12], after calling your function,nums should be[1, 3, 12, 0, 0].

Note:

1.You must do this in-place without making a copy of the array.

2.Minimize the total number of operations.

解题方法：

题意是把数组nums中0的元素后置，同时不能采用赋值数组。两种方法：

1.遇到是0的元素从数组最后向前存储并移位，遇到非0元素从前存储；

2.推荐：从前往后查找，不是0的元素前移，并计算0的个数，后面的全置0。

我的代码：

方法一：Runtime: **28 ms**

```
void moveZeroes(int* nums, int numsSize) {
    int endNum;         //从后计数0
    int startNum;       //从前计数非0
    int temp;
```

```
    int i,j;

    i = 0;
    startNum = 0;
    endNum = 0;
    while( (i+endNum) < numsSize ) {
        if(nums[i]==0) {
            //依次前移
            for(j=startNum; j<numsSize-endNum-1; j++) { //j少一个数
                nums[j] = nums[j+1];
            }
            nums[numsSize-endNum-1] = 0;
            endNum++;
        }
        else {
            nums[startNum] = nums[i];
            startNum++;
            i++;
        }
    }
}
```

方法二：Runtime: **8 ms**

```
void moveZeroes(int* nums, int numsSize) {
    int count;   //计算0的个数
    int i,j;
    int n;

    n = 0;
    count = 0;
    for(i=0; i<numsSize; i++) {
        if(nums[i]==0) {
            count++;
        }
        else  {
            nums[n] = nums[i];
            n++;
        }
    }
    //后置0
    for(j=0; j<count; j++) {
        nums[n] = 0;
        n++;
    }
}
```