283 Move Zeroes

Question:

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.

For example, given nums = [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.

来自 <https://leetcode.com/problems/move-zeroes/description/>

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

例如， 定义 nums = [0, 1, 0, 3, 12]，调用函数之后， nums 应为 [1, 3, 12, 0, 0]。

注意事项:

1. 必须在原数组上操作，不要为一个新数组分配额外空间。
2. 尽量减少操作总数。

Solution for Python3:

```python
class Solution1:
    def moveZeroes(self, nums):
        """
        :type nums: List[int]
        :rtype: void Do not return anything, modify nums in-place instead.
        """
        cnt = 0
        for i in range(len(nums)):
            if nums[i]:
                nums[cnt] = nums[i]
                if i > cnt:
                    nums[i] = 0
                cnt += 1

class Solution2:
    def moveZeroes(self, nums):
        """
        :type nums: List[int]
        :rtype: void Do not return anything, modify nums in-place instead.
        """
        cnt = 0
        for i in range(len(nums)):
            if nums[i]:
                nums[cnt] = nums[i]
                cnt += 1
        for i in range(cnt, len(nums)):
            nums[cnt] = 0
            cnt += 1

class Solution3:
    def moveZeroes(self, nums):
        """
```

```python
        :type nums: List[int]
        :rtype: void Do not return anything, modify nums in-place instead.
        """
        cnt = 0
        for i in range(len(nums)):
            if nums[i]:
                nums[cnt], nums[i] = nums[i], nums[cnt]
                cnt += 1
```

Solution for C++:

```cpp
class Solution1 {
public:
    void moveZeroes(vector<int>& nums) {
        int cnt = 0;
        for (int i = 0; i < nums.size(); i++) {
            if (nums[i]) {
                nums[cnt] = nums[i];
                if (i > cnt) {
                    nums[i] = 0;
                }
                cnt++;
            }
        }
    }
};

class Solution2 {
public:
    void moveZeroes(vector<int>& nums) {
        int cnt = 0;
        for (int i = 0; i < nums.size(); i++) {
            if (nums[i])
                nums[cnt++] = nums[i];
        }
        for (int i = cnt; i < nums.size(); i++) {
            nums[i] = 0;
        }
    }
};

class Solution3 {
public:
    void moveZeroes(vector<int>& nums) {
        for (int cnt = 0, i = 0; i < nums.size(); i++) {
            if (nums[i]) {
                swap(nums[cnt++], nums[i]);
            }
        }
    }
};
```