Move Zeroes

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.

Credits:
Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Solution 1

```
// Shift non-zero values as far forward as possible
// Fill remaining space with zeros

public void moveZeroes(int[] nums) {
    if (nums == null || nums.length == 0) return;

    int insertPos = 0;
    for (int num: nums) {
        if (num != 0) nums[insertPos++] = num;
    }

    while (insertPos < nums.length) {
        nums[insertPos++] = 0;
    }
}
```

written by Kurteck original link here

Solution 2

```
void moveZeroes(vector<int>& nums) {
    int last = 0, cur = 0;

    while(cur < nums.size()) {
        if(nums[cur] != 0) {
            swap(nums[last], nums[cur]);
            last++;
        }

        cur++;
    }
}
```

written by jaewoo original link here

Solution 3

```
class Solution {
public:
    void moveZeroes(vector<int>& nums) {
        int j = 0;
        // move all the nonzero elements advance
        for (int i = 0; i < nums.size(); i++) {
            if (nums[i] != 0) {
                nums[j++] = nums[i];
            }
        }
        for (;j < nums.size(); j++) {
            nums[j] = 0;
        }
    }
};
```

written by hyzhang original link here