Given an array nums with n objects colored red, white, or blue, sort them **in-place** so that objects of the same color are adjacent, with the colors in the order red, white, and blue.

We will use the integers 0, 1, and 2 to represent the color red, white, and blue, respectively.

You must solve this problem without using the library's sort function.

Example 1:

Input: nums = [2,0,2,1,1,0]
Output: [0,0,1,1,2,2]

Example 2:

Input: nums = [2,0,1]
Output: [0,1,2]

Constraints:

- n == nums.length
- 1 <= n <= 300
- nums[i] is either 0, 1, or 2.

Follow up: Could you come up with a one-pass algorithm using only constant extra space?

Approach:
The provided code sorts an array of integers nums that only contains the values 0, 1, and 2, commonly known as the Dutch National Flag problem.

Code:

```
class Solution {
    public void sortColors(int[] nums) {
        int []count=new int[3];
        for(int i=0;i<nums.length;i++){
            count[nums[i]]++;
        }
        int j=0;
        for(int i=0;i<count.length;i++){
            while(count[i]>0){
                nums[j]=i;
                j++;count[i]--;
            }
        }
    }
}
```