

# 075. Sort Colors

## 075 Sort Colors

### Description

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

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

Note:

You are not suppose to use the library's sort function for this problem.

### Solution

- Array
- Sort
- Two Pointers
- ...

#### Sort

- Count Sort

```
1 class Solution {
2 public:
3     void sortColors(vector<int>& nums) {
4         vector<int> colourNo(3);
5         for (int i=0; !nums.empty() && i<=nums.size()-1; ++i)
6             ++colourNo[nums[i]];
7
8         int spot = -1;
9         for (int i=0; !colourNo.empty() && i<=colourNo.size()-1; ++i){
10             int colour = i;
11             int colNo = colourNo[i];
12             for (int j=1; j<=colNo; ++j){
13                 ++spot;
14                 nums[spot] = i;
15             }
16         }
17         return;
18     }
19 };
```

#### Two Pointers

```
1 class Solution {
2 public:
3     void sortColors(vector<int>& nums) {
4         if(nums.empty() || nums.size()==1) return;
5
6         int redIndex = 0, blueIndex = nums.size()-1;
7         for (int i=0; i<=blueIndex;){
8             if (nums[i]==0){
9                 swap(nums[i], nums[redIndex]);
10                redIndex++;
11                ++i;
12            }
13            else if (nums[i]==2){
14                swap(nums[i], nums[blueIndex]);
15                blueIndex--;
```

```
16     }
17     else if (nums[i]==1)
18         i++;
19     }
20     return;
21 }
22 };
23
```