75. Sort Colors

Difficulty	medium
☑	☑
Finished	@July 8, 2023
Problem	array
Previously asked company	Facebook Salesforce
website	leetcode

Question:

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]
```

Optimal solution:

We can solve this problem with inbuilt sort function or quick sort or merge sort with O(nlogn) time complexity. Or we can also do bucket sort for this case the time complexity is O(2n) technically it is O(n) but we need to do traversal two times.

We can solve this problem using 3 pointers with single pass with

Time complexity: O(n)

Space complexity: O(1)

```
class Solution(object):
    def sortColors(self, nums):
        l, r = 0, len(nums)-1
        i = 0

        def swap(a,b):
            temp = nums[a]
            nums[a] = nums[b]
            nums[b] = temp
```

```
        while i <= r:
            if nums[i] == 0:
                swap(i,l)
                l += 1
                i += 1
            elif nums[i] == 2:
                swap(i,r)
                r -= 1
            else:
                i += 1
```