Problem 75: Sort Colors

Problem Information

Difficulty: Medium
Acceptance Rate: 68.62%
Paid Only: No
Tags: Array, Two Pointers, Sorting

Problem Description

Given an array `nums` with `n` objects colored red, white, or blue, sort them **[in-place](https://en.wikipedia.org/wiki/In-place_algorithm) **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?

Code Snippets

C++:

```cpp
class Solution {
public:
void sortColors(vector<int>& nums) {

}
};
```

Java:

```java
class Solution {
public void sortColors(int[] nums) {

}
}
```

Python3:

```python
class Solution:
def sortColors(self, nums: List[int]) -> None:
"""
Do not return anything, modify nums in-place instead.
"""
```