

2826. Sorting Three Groups

Description

You are given a **0-indexed** integer array `nums` of length `n`.

The numbers from `0` to `n - 1` are divided into three groups numbered from `1` to `3`, where number `i` belongs to group `nums[i]`. Notice that some groups may be **empty**.

You are allowed to perform this operation any number of times:

- Pick number `x` and change its group. More formally, change `nums[x]` to any number from `1` to `3`.

A new array `res` is constructed using the following procedure:

1. Sort the numbers in each group independently.
2. Append the elements of groups `1`, `2`, and `3` to `res` **in this order**.

Array `nums` is called a **beautiful array** if the constructed array `res` is sorted in **non-decreasing** order.

Return *the minimum number of operations to make `nums` a beautiful array*.

Example 1:

```
Input: nums = [2,1,3,2,1]
Output: 3
Explanation: It's optimal to perform three operations:
1. change nums[0] to 1.
2. change nums[2] to 1.
3. change nums[3] to 1.
After performing the operations and sorting the numbers in each group, group 1 becomes equal to [0,1,2,3,4] and group 2 and group 3 become empty.
Hence, res is equal to [0,1,2,3,4] which is sorted in non-decreasing order.
It can be proven that there is no valid sequence of less than three operations.
```

Example 2:

```
Input: nums = [1,3,2,1,3,3]
Output: 2
Explanation: It's optimal to perform two operations:
1. change nums[1] to 1.
2. change nums[2] to 1.
After performing the operations and sorting the numbers in each group, group 1 becomes equal to [0,1,2,3], group 2 becomes empty, and group 3 becomes equal to [4,5]. Hence, res is equal to [0,1,2,3,4,5] which is sorted in non-decreasing order.
It can be proven that there is no valid sequence of less than two operations.
```

Example 3:

```
Input: nums = [2,2,2,2,3,3]
Output: 0
Explanation: It's optimal to not perform operations.
After sorting the numbers in each group, group 1 becomes empty, group 2 becomes equal to [0,1,2,3] and group 3 becomes equal to [4,5]. Hence, res is equal to [0,1,2,3,4,5] which is sorted in non-decreasing order.
```

Constraints:

- `1 <= nums.length <= 100`
- `1 <= nums[i] <= 3`

