Problem 3467: Transform Array by Parity

Problem Information

Difficulty: Easy
Acceptance Rate: 89.58%
Paid Only: No
Tags: Array, Sorting, Counting

Problem Description

You are given an integer array `nums`. Transform `nums` by performing the following operations in the **exact** order specified:

1. Replace each even number with 0. 2. Replace each odd numbers with 1. 3. Sort the modified array in **non-decreasing** order.

Return the resulting array after performing these operations.

Example 1:

Input: nums = [4,3,2,1]

Output: [0,0,1,1]

Explanation:

* Replace the even numbers (4 and 2) with 0 and the odd numbers (3 and 1) with 1. Now, `nums = [0, 1, 0, 1]`. * After sorting `nums` in non-descending order, `nums = [0, 0, 1, 1]`.

Example 2:

Input: nums = [1,5,1,4,2]

Output: [0,0,1,1,1]

Explanation:

* Replace the even numbers (4 and 2) with 0 and the odd numbers (1, 5 and 1) with 1. Now, `nums = [1, 1, 1, 0, 0]`. * After sorting `nums` in non-descending order, `nums = [0, 0, 1, 1, 1]`.

Constraints:

* `1 <= nums.length <= 100` * `1 <= nums[i] <= 1000`

Code Snippets

C++:

```
class Solution {
public:
vector<int> transformArray(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int[] transformArray(int[] nums) {

}
}
```

Python3:

```
class Solution:
def transformArray(self, nums: List[int]) -> List[int]:
```