Problem 2592: Maximize Greatness of an Array

Problem Information

Difficulty: Medium
Acceptance Rate: 60.71%
Paid Only: No
Tags: Array, Two Pointers, Greedy, Sorting

Problem Description

You are given a 0-indexed integer array `nums`. You are allowed to permute `nums` into a new array `perm` of your choosing.

We define the **greatness** of `nums` be the number of indices `0 <= i < nums.length` for which `perm[i] > nums[i]`.

Return _the**maximum** possible greatness you can achieve after permuting_ `nums`.

Example 1:

Input: nums = [1,3,5,2,1,3,1] **Output:** 4 **Explanation:** One of the optimal rearrangements is perm = [2,5,1,3,3,1,1]. At indices = 0, 1, 3, and 4, perm[i] > nums[i]. Hence, we return 4.

Example 2:

Input: nums = [1,2,3,4] **Output:** 3 **Explanation:** We can prove the optimal perm is [2,3,4,1]. At indices = 0, 1, and 2, perm[i] > nums[i]. Hence, we return 3.

Constraints:

* `1 <= nums.length <= 105` * `0 <= nums[i] <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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