Problem 2856: Minimum Array Length After Pair Removals

Problem Information

Difficulty: Medium
Acceptance Rate: 26.30%
Paid Only: No
Tags: Array, Hash Table, Two Pointers, Binary Search, Greedy, Counting

Problem Description

Given an integer array `num` sorted in non-decreasing order.

You can perform the following operation any number of times:

* Choose **two** indices, `i` and `j`, where `nums[i] < nums[j]`. * Then, remove the elements at indices `i` and `j` from `nums`. The remaining elements retain their original order, and the array is re-indexed.

Return the **minimum** length of `nums` after applying the operation zero or more times.

Example 1:

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

Output: 0

Explanation:

![](https://assets.leetcode.com/uploads/2024/05/18/tcase1.gif)

Example 2:

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

Output: 0

Explanation:

![](https://assets.leetcode.com/uploads/2024/05/19/tcase2.gif)

Example 3:

Input: nums = [1000000000,1000000000]

Output: 2

Explanation:

Since both numbers are equal, they cannot be removed.

Example 4:

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

Output: 1

Explanation:

![](https://assets.leetcode.com/uploads/2024/05/19/tcase3.gif)

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 109` * `nums` is sorted in **non-decreasing** order.

Code Snippets

C++:

```
class Solution {
public:
int minLengthAfterRemovals(vector<int>& nums) {
}
};
```

Java:

```
class Solution {
public int minLengthAfterRemovals(List<Integer> nums) {

}
}
```

Python3:

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