Problem 594: Longest Harmonious Subsequence

Problem Information

Difficulty: Easy
Acceptance Rate: 64.11%
Paid Only: No
Tags: Array, Hash Table, Sliding Window, Sorting, Counting

Problem Description

We define a harmonious array as an array where the difference between its maximum value and its minimum value is **exactly** `1`.

Given an integer array `nums`, return the length of its longest harmonious subsequence among all its possible subsequences.

Example 1:

Input: nums = [1,3,2,2,5,2,3,7]

Output: 5

Explanation:

The longest harmonious subsequence is `[3,2,2,2,3]`.

Example 2:

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

Output: 2

Explanation:

The longest harmonious subsequences are `[1,2]`, `[2,3]`, and `[3,4]`, all of which have a length of 2.

Example 3:

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

Output: 0

Explanation:

No harmonic subsequence exists.

Constraints:

* `1 <= nums.length <= 2 * 104` * `-109 <= nums[i] <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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