594 Longest Harmonious Subsequence (link)

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*.

A **subsequence** of array is a sequence that can be derived from the array by deleting some or no elements without changing the order of the remaining elements.

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
```

Example 3:

```
Input: nums = [1,1,1,1]
Output: 0
```

Constraints:

- `1 <= nums.length <= 2 * 10^4`
- `-10^9 <= nums[i] <= 10^9`

(scroll down for solution)

Solution

Language: cpp

Status: Accepted

```cpp
#include <vector>
#include <unordered_map>

using namespace std;

class Solution {
public:
    int findLHS(vector<int>& nums) {
        unordered_map<int, int> freq;

        // Подсчет частот каждого числа в массиве

        for (int num : nums) {
            freq[num]++;
        }

        int max_length = 0;

        // Проверка для каждого числа, есть ли его "гармоничная пара"

        for (auto& pair : freq) {
            int num = pair.first;
            if (freq.find(num + 1) != freq.end()) {
                max_length = max(max_length, pair.second + freq[num + 1]);
            }
        }

        return max_length;
    }
};
```