Problem 3416: Subsequences with a Unique Middle Mode II

Problem Information

Difficulty: Hard
Acceptance Rate: 12.44%
Paid Only: Yes
Tags: Array, Hash Table, Math, Combinatorics

Problem Description

Given an integer array `nums`, find the number of subsequences of size 5 of `nums` with a **unique middle mode**.

Since the answer may be very large, return it **modulo** `109 + 7`.

A **mode** of a sequence of numbers is defined as the element that appears the **maximum** number of times in the sequence.

A sequence of numbers contains a**unique mode** if it has only one mode.

A sequence of numbers `seq` of size 5 contains a **unique middle mode** if the _middle element_ (`seq[2]`) is a **unique mode**.

Example 1:

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

Output: 6

Explanation:

`[1, 1, 1, 1, 1]` is the only subsequence of size 5 that can be formed from this list, and it has a unique middle mode of 1.

Example 2:

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

Output: 4

Explanation:

`[1, 2, 2, 3, 4]` and `[1, 2, 3, 3, 4]` have unique middle modes because the number at index 2 has the greatest frequency in the subsequence. `[1, 2, 2, 3, 3]` does not have a unique middle mode because 2 and 3 both appear twice in the subsequence.

Example 3:

Input: nums = [0,1,2,3,4,5,6,7,8]

Output: 0

Explanation:

There does not exist a subsequence of length 5 with a unique middle mode.

Constraints:

* `5 <= nums.length <= 105` * `-109 <= nums[i] <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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