Problem 3583: Count Special Triplets

Problem Information

Difficulty: Medium
Acceptance Rate: 37.25%
Paid Only: No
Tags: Array, Hash Table, Counting

Problem Description

You are given an integer array `nums`.

A **special triplet** is defined as a triplet of indices `(i, j, k)` such that:

* `0 <= i < j < k < n`, where `n = nums.length` * `nums[i] == nums[j] * 2` * `nums[k] == nums[j] * 2`

Return the total number of **special triplets** in the array.

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

Example 1:

Input: nums = [6,3,6]

Output: 1

Explanation:

The only special triplet is `(i, j, k) = (0, 1, 2)`, where:

* `nums[0] = 6`, `nums[1] = 3`, `nums[2] = 6` * `nums[0] = nums[1] * 2 = 3 * 2 = 6` * `nums[2] = nums[1] * 2 = 3 * 2 = 6`

Example 2:

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

Output: 1

Explanation:

The only special triplet is `(i, j, k) = (0, 2, 3)`, where:

* `nums[0] = 0`, `nums[2] = 0`, `nums[3] = 0` * `nums[0] = nums[2] * 2 = 0 * 2 = 0` * `nums[3] = nums[2] * 2 = 0 * 2 = 0`

Example 3:

Input: nums = [8,4,2,8,4]

Output: 2

Explanation:

There are exactly two special triplets:

* `(i, j, k) = (0, 1, 3)` * `nums[0] = 8`, `nums[1] = 4`, `nums[3] = 8` * `nums[0] = nums[1] * 2 = 4 * 2 = 8` * `nums[3] = nums[1] * 2 = 4 * 2 = 8` * `(i, j, k) = (1, 2, 4)` * `nums[1] = 4`, `nums[2] = 2`, `nums[4] = 4` * `nums[1] = nums[2] * 2 = 2 * 2 = 4` * `nums[4] = nums[2] * 2 = 2 * 2 = 4`

Constraints:

* `3 <= n == nums.length <= 105` * `0 <= nums[i] <= 105`

Code Snippets

C++:

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

}
```

```
};
```

Java:

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

}
}
```

Python3:

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