Problem 2198: Number of Single Divisor Triplets

Problem Information

Difficulty: Medium
Acceptance Rate: 54.85%
Paid Only: Yes
Tags: Math

Problem Description

You are given a **0-indexed** array of positive integers `nums`. A triplet of three **distinct** indices `(i, j, k)` is called a **single divisor triplet** of `nums` if `nums[i] + nums[j] + nums[k]` is divisible by **exactly one** of `nums[i]`, `nums[j]`, or `nums[k]`.

Return _the number of**single divisor triplets** of _`nums` _._

Example 1:

Input: nums = [4,6,7,3,2] **Output:** 12 **Explanation:** The triplets (0, 3, 4), (0, 4, 3), (3, 0, 4), (3, 4, 0), (4, 0, 3), and (4, 3, 0) have the values of [4, 3, 2] (or a permutation of [4, 3, 2]). 4 + 3 + 2 = 9 which is only divisible by 3, so all such triplets are single divisor triplets. The triplets (0, 2, 3), (0, 3, 2), (2, 0, 3), (2, 3, 0), (3, 0, 2), and (3, 2, 0) have the values of [4, 7, 3] (or a permutation of [4, 7, 3]). 4 + 7 + 3 = 14 which is only divisible by 7, so all such triplets are single divisor triplets. There are 12 single divisor triplets in total.

Example 2:

Input: nums = [1,2,2] **Output:** 6 **Explanation:** The triplets (0, 1, 2), (0, 2, 1), (1, 0, 2), (1, 2, 0), (2, 0, 1), and (2, 1, 0) have the values of [1, 2, 2] (or a permutation of [1, 2, 2]). 1 + 2 + 2 = 5 which is only divisible by 1, so all such triplets are single divisor triplets. There are 6 single divisor triplets in total.

Example 3:

Input: nums = [1,1,1] **Output:** 0 **Explanation:** There are no single divisor triplets. Note that (0, 1, 2) is not a single divisor triplet because nums[0] + nums[1] + nums[2] = 3 and 3 is divisible by nums[0], nums[1], and nums[2].

Constraints:

* `3 <= nums.length <= 105` * `1 <= nums[i] <= 100`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public long singleDivisorTriplet(int[] nums) {

}
}
```

Python3:

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