Problem 2367: Number of Arithmetic Triplets

Problem Information

Difficulty: Easy
Acceptance Rate: 85.27%
Paid Only: No
Tags: Array, Hash Table, Two Pointers, Enumeration

Problem Description

You are given a **0-indexed** , **strictly increasing** integer array `nums` and a positive integer `diff`. A triplet `(i, j, k)` is an **arithmetic triplet** if the following conditions are met:

* `i < j < k`, * `nums[j] - nums[i] == diff`, and * `nums[k] - nums[j] == diff`.

Return _the number of unique**arithmetic triplets**._

Example 1:

Input: nums = [0,1,4,6,7,10], diff = 3 **Output:** 2 **Explanation:** (1, 2, 4) is an arithmetic triplet because both 7 - 4 == 3 and 4 - 1 == 3. (2, 4, 5) is an arithmetic triplet because both 10 - 7 == 3 and 7 - 4 == 3.

Example 2:

Input: nums = [4,5,6,7,8,9], diff = 2 **Output:** 2 **Explanation:** (0, 2, 4) is an arithmetic triplet because both 8 - 6 == 2 and 6 - 4 == 2. (1, 3, 5) is an arithmetic triplet because both 9 - 7 == 2 and 7 - 5 == 2.

Constraints:

* `3 <= nums.length <= 200` * `0 <= nums[i] <= 200` * `1 <= diff <= 50` * `nums` is **strictly** increasing.

Code Snippets

C++:

```
class Solution {
public:
int arithmeticTriplets(vector<int>& nums, int diff) {

}
};
```

Java:

```
class Solution {
public int arithmeticTriplets(int[] nums, int diff) {

}
}
```

Python3:

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