Problem 454: 4Sum II

Problem Information

Difficulty: Medium
Acceptance Rate: 57.74%
Paid Only: No
Tags: Array, Hash Table

Problem Description

Given four integer arrays `nums1`, `nums2`, `nums3`, and `nums4` all of length `n`, return the number of tuples `(i, j, k, l)` such that:

* `0 <= i, j, k, l < n` * `nums1[i] + nums2[j] + nums3[k] + nums4[l] == 0`

Example 1:

Input: nums1 = [1,2], nums2 = [-2,-1], nums3 = [-1,2], nums4 = [0,2] **Output:** 2 **Explanation:** The two tuples are: 1. (0, 0, 0, 1) -> nums1[0] + nums2[0] + nums3[0] + nums4[1] = 1 + (-2) + (-1) + 2 = 0 2. (1, 1, 0, 0) -> nums1[1] + nums2[1] + nums3[0] + nums4[0] = 2 + (-1) + (-1) + 0 = 0

Example 2:

Input: nums1 = [0], nums2 = [0], nums3 = [0], nums4 = [0] **Output:** 1

Constraints:

* `n == nums1.length` * `n == nums2.length` * `n == nums3.length` * `n == nums4.length` * `1 <= n <= 200` * `-228 <= nums1[i], nums2[i], nums3[i], nums4[i] <= 228`

Code Snippets

C++:

```
class Solution {
public:
int fourSumCount(vector<int>& nums1, vector<int>& nums2, vector<int>& nums3,
vector<int>& nums4) {

}
};
```

Java:

```
class Solution {
public int fourSumCount(int[] nums1, int[] nums2, int[] nums3, int[] nums4) {

}
}
```

Python3:

```
class Solution:
def fourSumCount(self, nums1: List[int], nums2: List[int], nums3: List[int],
nums4: List[int]) -> int:
```