Problem 2395: Find Subarrays With Equal Sum

Problem Information

Difficulty: Easy
Acceptance Rate: 66.72%
Paid Only: No
Tags: Array, Hash Table

Problem Description

Given a **0-indexed** integer array `nums`, determine whether there exist **two** subarrays of length `2` with **equal** sum. Note that the two subarrays must begin at **different** indices.

Return `true` _if these subarrays exist, and_`false` _otherwise._

A **subarray** is a contiguous non-empty sequence of elements within an array.

Example 1:

Input: nums = [4,2,4] **Output:** true **Explanation:** The subarrays with elements [4,2] and [2,4] have the same sum of 6.

Example 2:

Input: nums = [1,2,3,4,5] **Output:** false **Explanation:** No two subarrays of size 2 have the same sum.

Example 3:

Input: nums = [0,0,0] **Output:** true **Explanation:** The subarrays [nums[0],nums[1]] and [nums[1],nums[2]] have the same sum of 0. Note that even though the subarrays have the same content, the two subarrays are considered different because they are in different positions in the original array.

Constraints:

* `2 <= nums.length <= 1000` * `-109 <= nums[i] <= 109`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public boolean findSubarrays(int[] nums) {

}
}
```

Python3:

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