<https://leetcode.com/problems/find-subarrays-with-equal-sum/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

-10^9 <= nums[i] <= 10^9

**Attempt 1: 2024-01-27**

**Solution 1: Hash Table (10 min)**

**Style 1: HashMap**

class Solution {

    public boolean findSubarrays(int[] nums) {

        Map<Integer, Integer> map = new HashMap<>();

        for(int i = 0; i < nums.length - 1; i++) {

            int sum = nums[i] + nums[i + 1];

            if(map.containsKey(sum)) {

                return true;

            }

            map.put(sum, map.getOrDefault(sum, 0) + 1);

        }

        return false;

    }

}

Time Complexity: O(N)

Space Complexity: O(N)

**Style 2: HashSet**

class Solution {

    public boolean findSubarrays(int[] nums) {

        Set<Integer> set = new HashSet<>();

        for(int i = 0; i < nums.length - 1; i++) {

            int sum = nums[i] + nums[i + 1];

            if(set.contains(sum)) {

                return true;

            }

            set.add(sum);

        }

        return false;

    }

}

Time Complexity: O(N)

Space Complexity: O(N)