**[Binary Subarrays With Sum](https://leetcode.com/problems/binary-subarrays-with-sum/)**

Given a binary array nums and an integer goal, return *the number of non-empty****subarrays****with a sum* goal.

A **subarray** is a contiguous part of the array.

**Example 1:**

**Input:** nums = [1,0,1,0,1], goal = 2

**Output:** 4

**Explanation:** The 4 subarrays are bolded and underlined below:

[**1,0,1**,0,1]

[**1,0,1,0**,1]

[1,**0,1,0,1**]

[1,0,**1,0,1**]

**Example 2:**

**Input:** nums = [0,0,0,0,0], goal = 0

**Output:** 15

**Constraints:**

* 1 <= nums.length <= 3 \* 104
* nums[i] is either 0 or 1.
* 0 <= goal <= nums.length

class Solution {

public:

    int numSubarraysWithSum(vector<int>& nums, int goal) {

        unordered\_map<int, int> count;

        count[0] = 1;

        int curr\_sum = 0;

        int total\_subarrays = 0;

        for (int num : nums) {

            curr\_sum += num;

            if (count.find(curr\_sum - goal) != count.end()) {

                total\_subarrays += count[curr\_sum - goal];

            }

            count[curr\_sum]++;

        }

        return total\_subarrays;

    }

};

Link : <https://leetcode.com/problems/binary-subarrays-with-sum/?envType=daily-question&envId=2024-03-14>