# 题目

给你一个数组 nums 。数组「动态和」的计算公式为：runningSum[i] = sum(nums[0]…nums[i]) 。

请返回 nums 的动态和。

示例 1：

输入：nums = [1,2,3,4]

输出：[1,3,6,10]

解释：动态和计算过程为 [1, 1+2, 1+2+3, 1+2+3+4] 。

示例 2：

输入：nums = [1,1,1,1,1]

输出：[1,2,3,4,5]

解释：动态和计算过程为 [1, 1+1, 1+1+1, 1+1+1+1, 1+1+1+1+1] 。

示例 3：

输入：nums = [3,1,2,10,1]

输出：[3,4,6,16,17]

提示：

1 <= nums.length <= 1000

-10^6 <= nums[i] <= 10^6

# 分析

## 方法一：原地修改

代码：

class Solution {

public:

vector<int> runningSum(vector<int>& nums) {

int n = nums.size();

for (int i = 1; i < n; i++) {

nums[i] += nums[i - 1];

}

return nums;

}

};

**复杂度：**

时间复杂度：O(n)，其中n是给定数组长度。

空间复杂度：O(1)。我们只需要常数的空间保存若干变量。