# 题目

给你一个整数数组 nums 。你可以选定任意的 正数 startValue 作为初始值。

你需要从左到右遍历 nums 数组，并将 startValue 依次累加上 nums 数组中的值。

请你在确保累加和始终大于等于 1 的前提下，选出一个最小的 正数 作为 startValue。

示例 1：

输入：nums = [-3,2,-3,4,2]

输出：5

解释：如果你选择 startValue = 4，在第三次累加时，和小于 1 。

累加求和

  startValue = 4 | startValue = 5 | nums

  (4 -3 ) = 1 | (5 -3 ) = 2 | -3

  (1 +2 ) = 3 | (2 +2 ) = 4 | 2

  (3 -3 ) = 0 | (4 -3 ) = 1 | -3

  (0 +4 ) = 4 | (1 +4 ) = 5 | 4

  (4 +2 ) = 6 | (5 +2 ) = 7 | 2

示例 2：

输入：nums = [1,2]

输出：1

解释：最小的 startValue 需要是正数。

示例 3：

输入：nums = [1,-2,-3]

输出：5

提示：

1 <= nums.length <= 100

-100 <= nums[i] <= 100

# 分析

class Solution {

public:

int minStartValue(vector<int>& nums) {

int total = 0, startValue = 1;

for(int i = 0; i < nums.size(); i++){

total += nums[i];

if(total < startValue) startValue = total;

}

return startValue < 1 ? (1 - startValue) : 1;

}

};

另一种写法：

int minStartValue(vector<int>& nums) {

int nmin = INT\_MAX, cur = 0;

for (int num : nums)

{

cur += num;

nmin = min(nmin, cur);

}

return max(1, 1 - nmin);

}

或：

class Solution {

public:

int minStartValue(vector<int>& nums) {

int val=1, i, sum=0;

for(i = 0; i < nums.size(); ++i)

{

sum += nums[i];

if(sum >= 1)

continue;

else

val = max(val, 1-sum);

}

return val;

}

};