# 题目

给你一个整数数组nums。数组中唯一元素是那些只出现恰好一次的元素。

请你返回nums中唯一元素的和。

示例 1：

输入：nums = [1,2,3,2]

输出：4

解释：唯一元素为[1,3]，和为4。

示例 2：

输入：nums = [1,1,1,1,1]

输出：0

解释：没有唯一元素，和为0。

示例 3 ：

输入：nums = [1,2,3,4,5]

输出：15

解释：唯一元素为[1,2,3,4,5]，和为15。

提示：

1 <= nums.length <= 100

1 <= nums[i] <= 100

# 分析

## 方法一：哈希表

思路：

通过哈希表来解决。遍历数组 nums，统计每个数字出现的次数，并将结果存储在哈希表中。然后再次遍历哈希表，找出只出现一次的数字，并将它们的和累加起来。

代码：

class Solution {

public:

int sumOfUnique(vector<int>& nums) {

std::unordered\_map<int, int> count;

int sum = 0;

for (int num : nums) {

count[num]++;

}

for (auto& p : count) {

if (p.second == 1) {

sum += p.first;

}

}

return sum;

}

};

或：

class Solution {

public:

int sumOfUnique(vector<int>& nums) {

std::unordered\_map<int, int> count;

int sum = 0;

for (int num : nums) {

count[num]++;

if (count[num] == 1) {

sum += num;

} else if (count[num] == 2) {

sum -= num;

}

}

return sum;

}

};