# 题目

给你一个整数数组 salary ，数组里每个数都是 唯一 的，其中 salary[i] 是第 i 个员工的工资。

请你返回去掉最低工资和最高工资以后，剩下员工工资的平均值。

示例 1：

输入：salary = [4000,3000,1000,2000]

输出：2500.00000

解释：最低工资和最高工资分别是 1000 和 4000 。

去掉最低工资和最高工资以后的平均工资是 (2000+3000)/2= 2500

示例 2：

输入：salary = [1000,2000,3000]

输出：2000.00000

解释：最低工资和最高工资分别是 1000 和 3000 。

去掉最低工资和最高工资以后的平均工资是 (2000)/1= 2000

示例 3：

输入：salary = [6000,5000,4000,3000,2000,1000]

输出：3500.00000

示例 4：

输入：salary = [8000,9000,2000,3000,6000,1000]

输出：4750.00000

提示：

3 <= salary.length <= 100

10^3 <= salary[i] <= 10^6

salary[i] 是唯一的。

与真实值误差在 10^-5 以内的结果都将视为正确答案。

# 分析

要解决“去掉最低工资和最高工资后计算平均工资”的问题，核心思路是先找到数组中的最大值和最小值，再计算剩余元素的总和与平均值，步骤清晰且效率高效。

解题思路：

1、关键观察：

题目要求去掉唯一的最低工资和最高工资（数组中所有元素唯一），因此只需先确定这两个极值，再从总工资中扣除，最后除以剩余元素的个数（总长度 - 2）。

2、步骤拆解：

- 计算总工资：遍历数组求和，或使用标准库函数快速求和。

- 找到极值：遍历数组找到最大值和最小值，或使用标准库函数直接获取。

- 计算平均值：用“总工资 - 最大值 - 最小值”得到剩余工资总和，再除以“数组长度 - 2”（剩余元素个数），结果需转为浮点型以保证精度。

代码：

class Solution {

public:

double average(vector<int>& salary) {

// 1. 计算工资总和

int total = accumulate(salary.begin(), salary.end(), 0);

// 2. 找到最大值和最小值

int max\_sal = \*max\_element(salary.begin(), salary.end());

int min\_sal = \*min\_element(salary.begin(), salary.end());

// 3. 计算剩余工资的平均值（转为double避免整数除法）

int n = salary.size();

return (double)(total - max\_sal - min\_sal) / (n - 2);

}

};

代码解释：

1、求和（accumulate）：

标准库函数accumulate从salary.begin()到salary.end()遍历数组，初始值为0，直接计算所有工资的总和，时间复杂度O(n)（n为数组长度）。

2、找极值（max\_element/min\_element）：

标准库函数max\_element和min\_element分别返回数组中最大值和最小值的迭代器，通过解引用（\*）获取具体值，时间复杂度均为O(n)。

3、计算平均值：

- 剩余工资总和 = 总工资 - 最大值 - 最小值。

- 剩余元素个数 = 数组长度 - 2（去掉1个最大值和1个最小值）。

- 强制将分子转为double，避免整数除法导致的精度丢失，最终返回浮点型结果。

复杂度分析

- 时间复杂度：O(n)，仅需遍历数组3次（求和1次，找最大值1次，找最小值1次），实际可优化为1次遍历（同时求和、找极值），但标准库函数已足够高效，且代码更简洁。

- 空间复杂度：O(1)，仅使用常数个变量（total、max\_sal、min\_sal、n），不依赖输入规模。