

P1223 排队接水

题目描述

有 n 个人在一个水龙头前排队接水，假如每个人接水的时间为 T_i ，请编程找出这 n 个人排队的一种顺序，使得 n 个人的平均等待时间最小。

输入格式

第一行为一个整数 n 。

第二行 n 个整数，第 i 个整数 T_i 表示第 i 个人的等待时间 T_i 。

输出格式

输出文件有两行，第一行为一种平均时间最短的排队顺序；第二行为这种排列方案下的平均等待时间（输出结果精确到小数点后两位）。

输入样例

```
10
56 12 1 99 1000 234 33 55 99 812
```

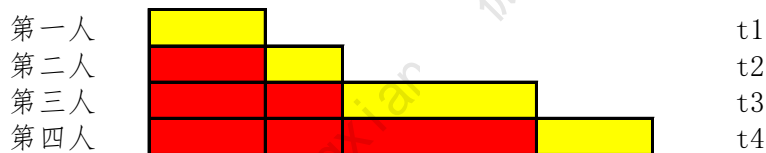
输出样例

```
3 2 7 8 1 4 9 6 10 5
291.90
```

解析

本题中有个隐藏信息：自己开始接水时，自己的消耗时间不计入总时间。

求最短平均时间就是求所有人的最短等待时间和。由于排队接水是一个接着一个的，也就是只允许最多一个人同时打水，所以某一个人打水的时候，其身后的人的等待时间总和就是每个单人打水时间的和。如下图所示：



其中黄色部分是每个人自己打水的时间，红色部分则是总共的等待时间。

根据图中的信息，我们可以发现 t_1 会被计算3次， t_2 会被计算2次， t_3 会被计算1， t_4 不会被计算。通过观察，我们感觉到，应该按照每个人接水的时间从小到大进行排列，这样才能保证总的等待和最小。

有了想法之后，我们需要对其进行证明。

假设最佳方案中， t_1 到 t_n 是从大到小排列。

即当 $i < j$ 时， $t_i > t_j$ ，这两项贡献的总时间是 $s_1 = a * t_i + b * t_j$ ，其中 $(a > b)$ 。

若将 t_i 和 t_j 调换，那么贡献总时间变为 $s_2 = a * t_j + b * t_i$ 。

两者相减 $s_1 - s_2 = a * (t_i - t_j) - b * (t_i - t_j) = (a - b) * (t_i - t_j) > 0$ 。

说明调换后，总得时间会缩短，这与原来的“最佳方案”矛盾，所以贪心算法成立。

编码

```
#include <bits/stdc++.h>

using namespace std;

const int MaxN = 1001;
//定义学生的结构体
//包括学生id和接水用时
struct Student {
    int id;
    int time;
};

int n;
double sum;
double avg;
Student students[MaxN];
```

```

//按照打水时间的长短进行排序，小的在前
bool Compare(Student a, Student b) {
    if (a.time != b.time) {
        return a.time < b.time;
    }
    //相等的时候按照id进行排序
    //注意控制稳定性
    return a.id < b.id;
}

void Print() {
    //打印每个学生的id
    for (int i = 1; i <= n; ++i) {
        printf("%d ", students[i].id);
    }
    cout << endl;
    //平均用水的时间
    printf("%.2lf", avg);
}

int main(int argc, char **argv) {
    scanf("%d", &n);
    //观察输出数据，可以得出是从1开始进行计数
    for (int i = 1; i <= n; ++i) {
        scanf("%d", &students[i].time);
        students[i].id = i;
    }
    //先按照接水的时间从小到大排序
    sort(students + 1, students + 1 + n, Compare);
    //计算总的等待时间
    for (int i = 1; i <= n; ++i) {
        //第一个人接水的时候，后面有九个人在等待，
        //第二个人接水的时候，后面有八个人在等待，
        //以此类推...
        sum += students[i].time * (n - i);
    }
    //计算平均值
    avg = sum * 1.0 / n;
    Print();
    return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

