



题解

——2023/03/04 石硯

A、相等的代价

D、立体最短路

Contents

1

A、相等的代价

2

D、立体最短路

A、相等的代价——题板

题目描述

给定一个整数序列 a_1, \dots, a_n

Mas 需要修改其中一些数字,将它调整成一个全部相等的整数序列

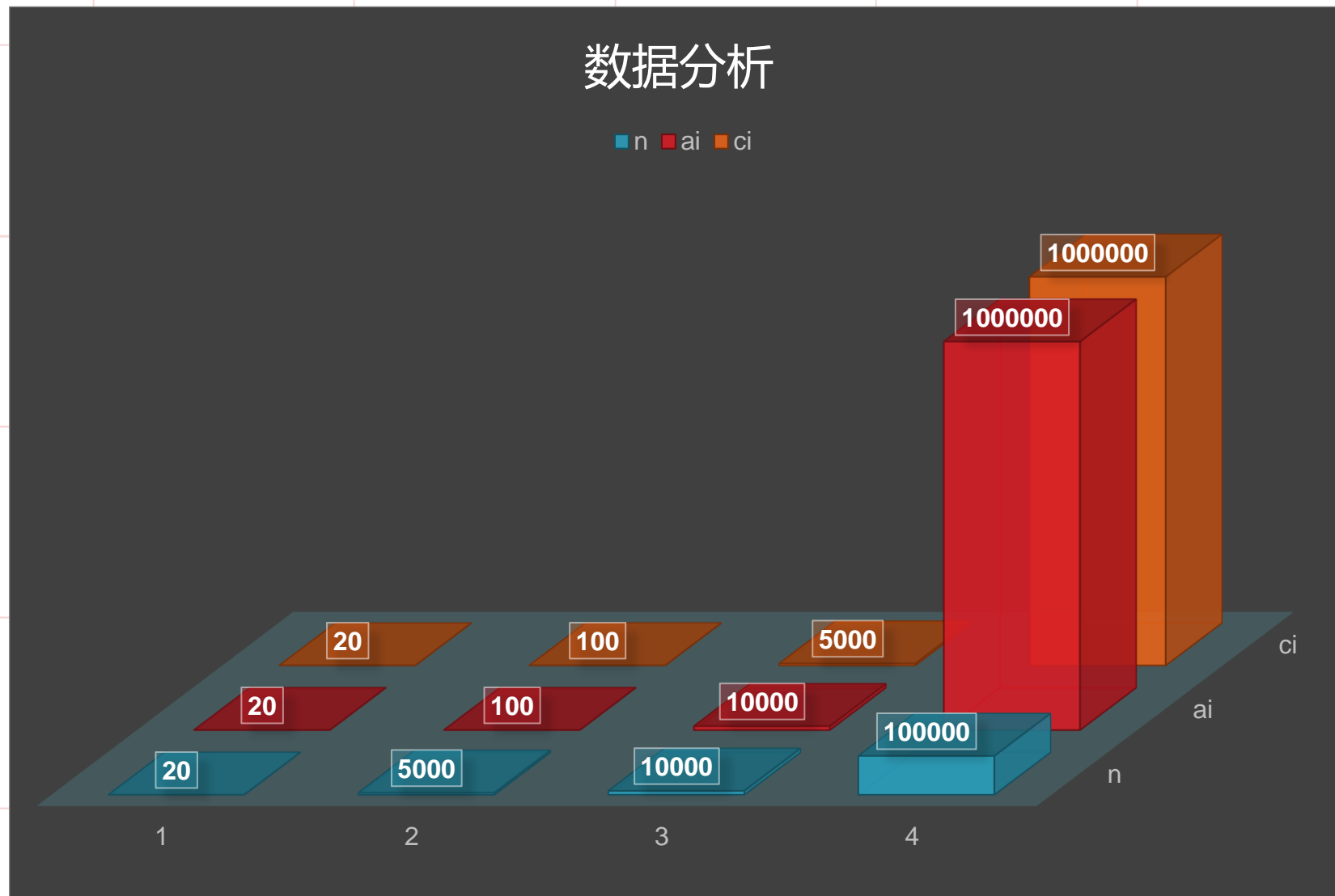
Mas 可以执行下面操作 **任意** 次:

将 a_i 增加或者减小 1

若 a_i 进行一次修改代价为 c_i

请找到一种修改方法,使得修改代价的总和达到最小,输出这个最小值

A、相等的代价——数据分析



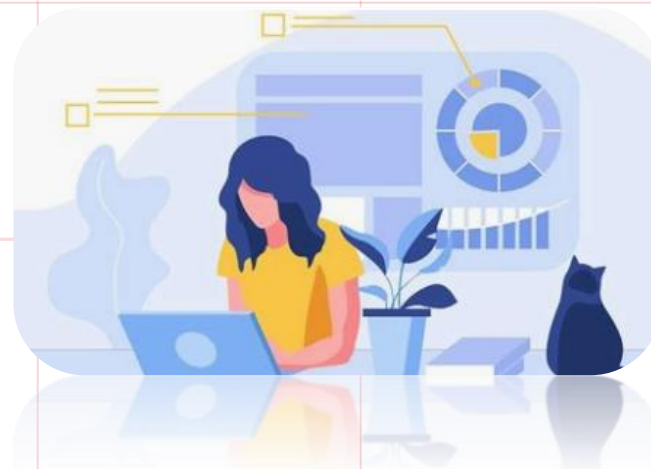
A、相等的代价——分析（思路1）



复制数据

取中位数

输出结果



A、相等的代价——分析

例如：1 2 3

2 1 3

会变成：1 1 2 3 3 3

其实，变成 a_3 a_4 都可以（因为都是中位数）

再以样例1为准：1 3 5 2

2 3 1 14

会变成：1 1 3 3 3 5 2 2 2 2 2 2 2 2 2 2 2 2 2 2

排序一下：1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 5

我们发现，中位数是2（上方标橙）

∴2就是答案

A、相等的代价——分析

然而，如果要把数据全部复制下来，共有 $10^5 \times 10^6 = 10^{11}$ 个元素（应该不会有哪一个dcm开一个 $a[10^{11}]$ 的数组吧）。

我们完全可以先按照 a_i 排序（定义一个struct），在做累加，判断

$$sum \text{ 是否 } \geq \left\lfloor \frac{\sum_{i=1}^n c_i}{2} \right\rfloor$$

如果本式成立，那么现在计算的 a_i 就是最优的基准数，计算即可。

```
struct node
{
    int data, c;
    bool operator<(const node T) const
    {
        return data < T.data;
    }
} a[SIZE];
```

```
sum = (sum + 1 >> 1);
for (int i = 1; i <= n; i++)
{
    sum2 += a[i].c;
    if (sum2 >= sum)
    {
        num = a[i].data;
        break;
    }
}
```

A、相等的代价——分析（思路2）



升序排序

推式子

输出结果



A、相等的代价——分析

将所有元素修改为 a_x 的代价为：

$$\sum_{i=1}^{x-1} (c_i \times (a_x - a_i)) + \sum_{i=x+1}^n (c_i \times (a_i - a_x))$$

可推得：

$$\sum_{i=1}^{x-1} (c_i \times a_x) - \sum_{i=1}^{x-1} (c_i \times a_i) + \sum_{i=x+1}^n (c_i \times a_i) - \sum_{i=x+1}^n (c_i \times a_x)$$

令 $sum_{1,i} = \sum_{j=1}^i (a_j \times c_j)$

$$sum_{2,i} = \sum_{j=1}^i c_j$$

A、相等的代价——代码

```
for (int i = 1; i <= n; i++)
{
    sum1[i] = sum1[i - 1] + a[i].data * a[i].t;
    sum2[i] = sum2[i - 1] + a[i].t;
}
for (int i = 1; i <= n; i++)
{
    long long now = a[i].data * sum2[i - 1] - sum1[i - 1] + (sum1[n] - sum1[i])
                    - a[i].data * (sum2[n] - sum2[i]);
    ans = min(now, ans);
}
```

D、立体最短路——题板

题目描述

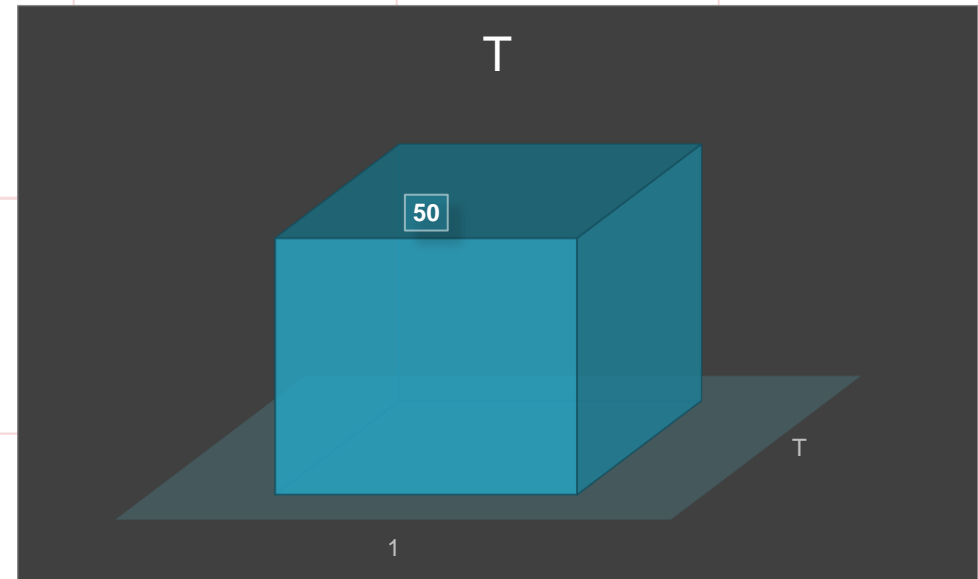
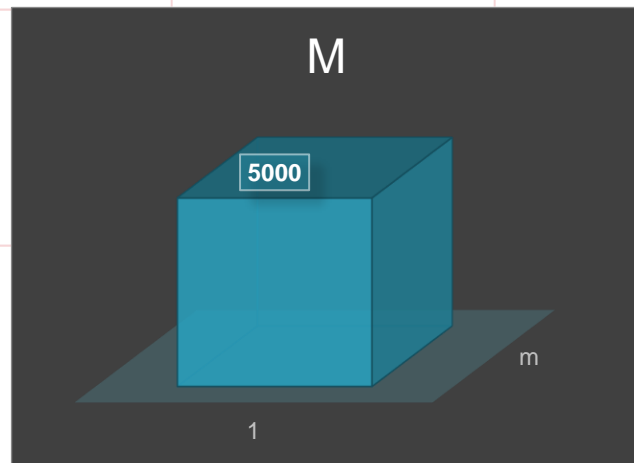
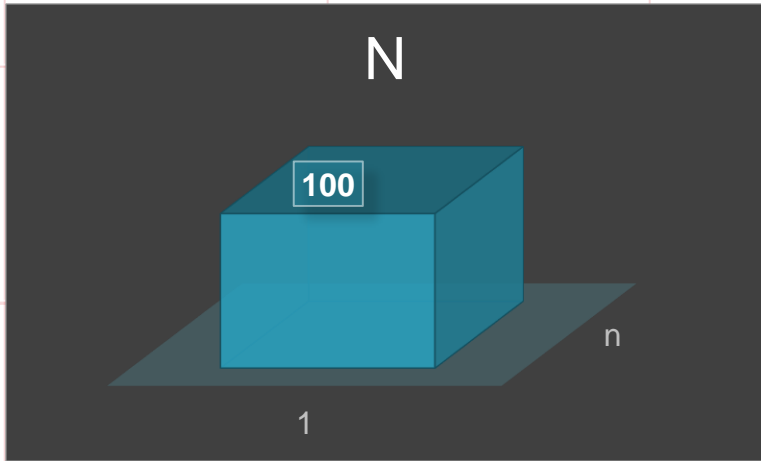
Mas 最近在玩一个塔防游戏,游戏处于一个 $n \times n \times n$ 个立方体构成的三维空间

在空间内 *Mas* 每次只能向前、向后、向左、向右、向上、向上移动一步

不幸的是,空间内存在 m 个障碍物,每个障碍物给定两个三维坐标 (x_1, y_1, z_1) 和 (x_2, y_2, z_2) , 对于一个立方体 (x, y, z) 如果 $x_1 \leq x \leq x_2, y_1 \leq y \leq y_2, z_1 \leq z \leq z_2$ 那么这个立方体就是障碍无法通过

现在给定 *Mas* 所在的起点 (sx, sy, sz) 和终点 (ex, ey, ez) , 你请求出最少步数

D、立体最短路——数据分析



D、立体最短路——分析

我们以单个平面开始。

$$\begin{cases} sx \\ sy \\ sz \end{cases}$$

$$\begin{cases} ex + 1 \\ sy \\ sz \end{cases} \quad (\text{ex要+1是因为三元组} \begin{cases} ex \\ sy \\ sz \end{cases} \text{也算})$$

显然，这边要-1

$$\begin{cases} sx \\ ey + 1 \\ sz \end{cases} \quad \text{显然，这边也要-1}$$

$$\begin{cases} ex + 1 \\ ey + 1 \\ sz \end{cases} \quad \text{显然，这边要+1}$$

D、立体最短路——分析

通过上面的分析不难看出，三维差分的式子：

```
d[ex + 1][ey + 1][ez + 1]--;  
d[sx][ey + 1][ez + 1]++;  
d[ex + 1][sy][ez + 1]++;  
d[ex + 1][ey + 1][sz]++;  
d[sx][sy][ez + 1]--;  
d[sx][ey + 1][sz]--;  
d[ex + 1][sy][sz]--;  
d[sx][sy][sz]++;
```

还原vis数组：

```
vis[i][j][k] = vis[i - 1][j][k] + vis[i][j - 1][k] +  
vis[i][j][k - 1] - vis[i - 1][j - 1][k] - vis[i - 1][j][k - 1] - vis[i][j - 1][k - 1] +  
vis[i - 1][j - 1][k - 1];  
d[i][j][k];
```

最后做一下bfs搜索就可以了

注意：vis要开int

还可以做一做

<https://oj.shiyancang.cn/Contest/Problem/469.html?index=3&cid=5270&type=4>

D、立体最短路——练习（星球大战）

题目描述

在公元 22022 年,宇宙中有个星球叫**达瓦里氏星**,这颗红色星球本来是宇宙中最强大的星球,可是后来被其他星球攻击,最后分裂若干个小星球,其中就包括**天鹅星**和**乌鸦星**

虽然这两颗星球都起源于同一个星球,但是两星球之间的关系却异常的差,两星球之间经常发生冲突与摩擦,**乌鸦星**总是联合其他星球挑衅**天鹅星**

终于,**天鹅星**最终忍无可忍,决定对**乌鸦星**发动攻击,开启星球大战

天鹅星的太空飞船摆出了特殊的阵型,用来对付**乌鸦星**: **天鹅星**派出了 $n \times m \times l$ 艘战舰,在太空中排成一个 n 层 m 行 l 列的立方体,其中,第 i 层第 j 行第 k 列的战舰(记为战舰 (i, j, k)) 的战斗力为 $a(i, j, k)$

为了应对危机,**乌鸦星**需要进行 q 轮估计,每轮需要计算出阵型中指定立方体区域中战舰的生命值总和,你能帮帮他吗?

由于数据较大,因此我们计算立方体区域的总和时 $\text{mod } 1000000007$ 后输出

输入格式

第一行包括三个数 n, m, l

第二行包括 $n \times m \times l$ 个数,其中第 $((i-1) \times m + (j-1)) \times l + (k-1) + 1$ 个数为 $a(i, j, k)$

接下来一行包括一个数 q ,表示询问次数

接下来包括 q 行,每一行都有 6 个数,依次是 $x_1, y_1, z_1, x_2, y_2, z_2$,表示一个立方体的两个对角的顶点 $(x_1, y_1, z_1), (x_2, y_2, z_2)$,这两个顶点可以确定一个立方体

输出格式

对于 q 次询问中的每一轮询问,输出一行一个整数表示询问的立方体区域中战舰的生命值总和

输入样例

```
1 2 3
1 2 3 4 5 6
2
1 1 1 1 2 2
1 2 1 1 2 3
```

输出样例

```
12
15
```

数据规模

对于 10% 的数据 $1 \leq q \leq 10$

对于 100% 的数据 $1 \leq n, m, l \leq 100, 1 \leq q \leq 10^5, 1 \leq x_1 \leq x_2 \leq n, 1 \leq y_1 \leq y_2 \leq m, 1 \leq z_1 \leq z_2 \leq l$

D、立体最短——代码

```
43     while (T--)
44     {
10 void bfs(int sx, int sy, int sz, int ex, int ey, int ez)
11 {
12     while (q.size())
13         q.pop();
14     q.push({sx, sy, sz, 0});
15     while (q.size())
16     {
17         auto cur = q.front();
18         q.pop();
19         if (cur.x == ex && cur.y == ey && cur.z == ez)
20         {
21             ans = cur.step;
22             flag = true;
23             return;
24         }
25         for (int i = 0; i < 6; i++)
26         {
27             int tx = cur.x + dir[i][0], ty = cur.y + dir[i][1], tz = cur.z + dir[i][2];
28             if (tx && ty && tz && tx <= n && ty <= n && tz <= n && !vis[tx][ty][tz])
29             {
30                 vis[tx][ty][tz] = true;
31                 q.push({tx, ty, tz, cur.step + 1});
32             }
33         }
34     }
35 }
67     printf("%d\n", flag ? ans : -1);
68 }
```