

NOIP2020 模拟赛

Fuyuki

2020 年 11 月 28 日

8:00 ~ 12:30

【题目概况】

题目名称	逆天改命	星命定轨	风神之诗	蹦蹦炸弹
题目类型	传统	传统	传统	传统
可执行文件名	determination	destiny	poem	bomb
输入文件名	determination.in	destiny.in	poem	bomb
输出文件名	determination.out	destiny.out	poem.out	bomb
时间限制	1s	1s	2s	1s
空间限制	512MB	512MB	512MB	512MB
测试点数目	25	20	20	20
测试点是否等分	是	是	是	是
结果比较方式	自定义校验器 spj	自定义校验器 spj	全文比较	全文比较

【提交源程序文件名】

对于 C++ 语言	change.cpp	destiny.cpp	poem.cpp	bomb.cpp
-----------	------------	-------------	----------	----------

【编译选项】

对于 C++ 语言	-lm -O2 -std=c++11
-----------	--------------------

【注意事项】

1. 考试时间为 4.5 小时，请合理安排做题顺序。
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 `main()` 的返回值必须是 `int`，值必须为 0
4. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
5. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。

逆天改命

题目背景

你是倒序开题的。
此时此刻，考试已经结束，一切都已尘埃落定。
但别失去希望。
保持你的决心。
这会为你带来逆天改命的力量。

题目描述

再次审视题面，你发现出题人并没有给定每道题的总分，这就是希望所在。

这场考试一共有 n 名选手参加，此时你已经知道考号为 i 的选手在三道题中分别得到了 $a_i\%$, $b_i\%$, $c_i\%$ 的分数（你的考号为 1）。

如果三道题的总分分别为 A, B, C ，那么考号为 i 的选手的最终得分等于 $A \times a_i\% + B \times b_i\% + C \times c_i\%$ 。

你想成为这场考试的第一名，这意味着所有人的最终得分都必须小于等于你的最终得分。

现在你拥有了逆天改命的能力，可以随意指定 A, B, C 的值，但你需要满足以下条件：

- 每道题的总分不能过高也不能过低， $100 \leq A, B, C \leq 10^8$ 。
- 第一题的难度比后两题要低得多，所以需要满足 $A \leq B$ 且 $A \leq C$ 。

请求出一组满足条件的 A, B, C 使得在这种分数分配下你可以成为第一名，或者判断这是不可能的。

输入格式

第一行一个整数 T ，表示数据组数。

对于每组数据：

第一行一个整数 n 参加这场考试的选手人数。

接下来 n 行，每行三个整数 a_i, b_i, c_i 表示考号为 i 的选手的得分情况。

输出格式

每组数据输出一行，共 T 行。

对于第 i 组数据，如果不可能成为第一名，输出 -1 ，否则输出三个实数表示一组满足题意的 A, B, C 。

注意：你需要保证 $0 \leq A, B, C \leq 10^8$ ，本题允许的最大绝对误差为 10^{-6} 。

样例输入

```
1
3
30 72 65
60 88 40
100 40 60
```

样例输出

110 215 290

样例解释

考号为 1 的选手的得分为 $30\% \times 110 + 72\% \times 215 + 65\% \times 290 = 376$ 。

考号为 2 的选手的得分为 $60\% \times 110 + 88\% \times 215 + 40\% \times 290 = 371.2$ 。

考号为 3 的选手的得分为 $100\% \times 110 + 70\% \times 215 + 65\% \times 290 = 370$ 。

数据规模及约定

对于均匀分布的 40% 的数据，保证所有人都在后两题中爆零。（即 $b_i = 0, c_i = 0$ 对 $1 \leq i \leq n$ 成立。）

对于均匀分布的另外 40% 的数据，保证所有人都在最后一题中爆零。（即 $c_i = 0$ 对 $1 \leq i \leq n$ 成立。）

对于前 20% 的数据，保证 $1 \leq n \leq 20$ 。

对于前 60% 的数据，保证 $1 \leq n \leq 300$ 。

对于 100% 的数据，保证 $1 \leq T \leq 10, 1 \leq n \leq 10^3, 0 \leq a_i, b_i, c_i \leq 100$ 。

星命定轨

题目背景

万物的生命与天上的星相是对应的，观测星辰的命运就能窥视人的命运。
但就像人无力改变星辰的运行一样，没有人能阻止生命在既定的轨迹上前行。
占星术士就是这样矛盾的存在，探视着无可改变的命运，除了无力感外什么都得不到。

题目描述

这片星空能划分成 n 行 m 列共 $n \times m$ 个星区，根据占星术士的手稿，你得知第 i 行第 j 列的星区的命运参数为 $a_{i,j}$ 。

若干星区的集合被称为星域，每一方星域都对应着一个生命。不同形状的星域对应着不同的生命，其中矩形形状的星域对应人类。一个星域是矩形形状的，当且仅当存在四个参数 lx, rx, ly, ry 使得这个星域包含满足 $lx \leq i \leq rx, ly \leq j \leq ry$ 的所有第 i 行第 j 列的星区。

人生每个状态的价值可以由星域内的所有星区计算得到：

“生”的参数等于这些星区命运参数的加和。

“老”的参数等于这些星区命运参数的按位或之和。

“病”的参数等于这些星区命运参数的按位与之和。

“死”的参数等于这些星区命运参数的按位异或之和。

你希望对“生老病死”分别求出这片星空对应的所有人类的参数之和对 998244353 取模后的结果。

输入格式

第一行数两个整数 n, m ，表示星空划分成了 n 行 m 列。

接下来 n 行每行 m 个整数，第 i 行第 j 个数表示 $a_{i,j}$ 。

输出格式

一行四个整数，依次表示所有人“生”的参数之和，“老”的参数之和，“病”的参数之和，“死”的参数之和。

样例输入 1

```
2 2
1 2
3 4
```

样例输出 1

```
40 11 36 32
```

样例解释 1

总共有 9 个对应人类的星域，用形如 (lx, ly, rx, ry) 的四元组表示的结果如下：

$(1, 1, 1, 1), (1, 1, 1, 2), (1, 1, 2, 1), (1, 1, 2, 2), (1, 2, 1, 2), (1, 2, 2, 2), (2, 1, 2, 1), (2, 1, 2, 2), (2, 2, 2, 2)$

其“生老病死”参数之和分别为：

生： $1 + 3 + 4 + 10 + 2 + 6 + 3 + 7 + 4 = 40$ 。

老： $1 + 0 + 1 + 0 + 2 + 0 + 3 + 0 + 4 = 11$ 。

病： $1 + 3 + 3 + 7 + 2 + 6 + 3 + 7 + 4 = 36$ 。

死： $1 + 3 + 2 + 4 + 2 + 6 + 3 + 7 + 4 = 32$ 。

样例输入 2

```
5 5
1 0 1 0 1
1 0 0 1 1
0 1 0 1 1
0 0 1 1 0
1 1 0 0 1
```

样例输出 2

```
654 26 208 116
```

数据规模及约定

如果你只能求出部分参数的和，也可以获得部分分数：

求出正确的“生”的参数可以获得这个测试点 20% 的分数。

求出正确的“老”的参数可以获得这个测试点 20% 的分数。

求出正确的“病”的参数可以获得这个测试点 20% 的分数。

求出正确的“死”的参数可以获得这个测试点 40% 的分数。

如果你不能正确的求出某个参数，请在对应位置输出 -1 以避免校验器产生未知的错误。

对于前 20% 的数据，保证 $n, m \leq 10$ 。

对于前 50% 的数据，保证 $n, m \leq 100$ 。

对于另外 10% 的数据，保证 $n = 1$ 。

对于另外 20% 的数据，保证 $a_{i,j} \in \{0, 1\}$

对于 100% 的数据，保证 $1 \leq n \times m \leq 10^5, 0 \leq a_{i,j} \leq 10^9$ 。

风神之诗

题目背景

他拂动琴弦，用神风吹散冰雪，劈开山峦。
 让新的蒙德成为自由之地吧，成为无人称王的国度。
 假以时日，应该会是很好的、很浪漫的城邦…
 「他也会希望生活在这里的地方吧」

题目描述

由西向东排列着 n 座高山，第 i 座山的高度为 a_i 。
 每天，神圣的西风会吹过一段连续的山，为其带去祝福与希望。
 但是并非每座山峰都能得到风神的祝福。如果在风吹来的方向存在一座山的高度不低于当前这座山，那么吹向这座山的风会被全部挡住。
 形式化地讲，如果西风吹过 $[l, r]$ 内的所有高山，那么第 i ($l \leq i \leq r$) 座山能得到风神的祝福当且仅当不存在一座山 j ($l \leq j < i$) 满足 $a_j \geq a_i$ 。
 西风在山地连续吹拂了 m 天，请求出每一天都有多少座高山得到了风神的祝福。

输入格式

因为输入量可能很大，部分输入数据将在选手程序中生成。
 第一行包含三个整数 n, m, sd 表示高山的数量，西风吹拂的天数以及随机种子。
 接下来一行 n 个整数，第 i 个整数表示 a_i 。
 生成其余输入数据的数据生成器的 C++ 代码如下：

```
unsigned sd;
int rd() { return (sd ^= sd << 13, sd ^= sd >> 17, sd ^= sd << 5) % n + 1; }
```

选手程序在读入 n, sd 后，将通过调用函数 `rd` 生成剩下的输入数据。

记第 i 次调用函数 `rd` 的返回值为 val_i ，第 i 天西风将吹过 $[\min(val_{2i-1}, val_{2i}), \max(val_{2i-1}, val_{2i})]$ 范围内的所有高山。

输出格式

记第 i 天得到风神祝福的高山数量为 ans_i 。
 请输出 $\sum_{i=1}^m ans_i * i$ 对 998244353 取模后的结果。

样例输入

```
10 6 402338762
9 1 15 24 19 42 26 42 88 54
```

样例输出

```
43
```

样例解释

$val = 8, 1, 9, 7, 3, 6, 9, 8, 2, 3, 6, 7 \dots$

以下加粗的山得到了风神的祝福。

第 1 天西风吹过的区间是 $[1, 8]$: **9 1 15 24** 19 **42** 26 42 共 4 座山得到风神的祝福。

第 2 天西风吹过的区间是 $[7, 9]$: **26 42 88** 共 3 座山得到风神的祝福。

第 3 天西风吹过的区间是 $[3, 6]$: **15 24** 19 **42** 共 3 座山得到风神的祝福。

第 4 天西风吹过的区间是 $[8, 9]$: **42 88** 共 2 座山得到风神的祝福。

第 5 天西风吹过的区间是 $[2, 3]$: **1 15** 共 2 座山得到风神的祝福。

第 6 天西风吹过的区间是 $[6, 7]$: **42 26** 共 1 座山得到风神的祝福。

答案为: $4 \times 1 + 3 \times 2 + 3 \times 3 + 2 \times 4 + 2 \times 5 + 1 \times 6 = 43$ 。

数据规模及约定

对于前 20% 的数据, 保证 $n, m \leq 10^3$ 。

对于前 40% 的数据, 保证 $n, m \leq 10^5$ 。

对于另外 10% 的数据, 保证 $a_i \leq 10$ 。

对于另外 20% 的数据, 保证 $m \leq 10^6$ 。

对于 100% 的数据, 保证 $1 \leq n \leq 10^6, 1 \leq m \leq 10^7, 1 \leq a_i, sd \leq 10^9$ 。

蹦蹦炸弹

题目背景

投出带来无限快乐的蹦蹦炸弹！

题目描述

蹦蹦炸弹超凡的威力在于其子母弹的结构：巨大坚固的外壳能让炸弹深入怪物群中，随后外壳自毁并产生一次巨大的爆炸。伴随着冲击波，大量的诡雷散布开来，范围更广的二次爆炸对敌人带来毁灭性的打击。

在炸弹内部，诡雷用绳索结结实实地扎成一束，这样在外壳自毁前不会因为内部的碰撞导致诡雷被提前引爆。

在此之前，为了增加安全性，在 n 颗诡雷之间总共增加过 m 次绳索。每次增加绳索的时候都选择了四个参数 $x, y, l, w (x \neq y, \max(x, y) + l - 1 \leq n)$ ，对所有 $0 \leq i < l$ ，在编号为 $x + i$ 和 $y + i$ 的诡雷之间连接了一条强度为 w 的绳索。

固定诡雷的绳索强度之和越小，炸弹爆炸的时候诡雷也会炸得越远。但是如果绳索不能将所有诡雷都绑成一束，诡雷就有提前引爆的危险，即任意两颗诡雷之间都必须通过绳索直接或间接连到一起。

现在可莉需要一款火力加强型的蹦蹦炸弹，为此她决定剪掉部分原有的绳索。

请求出在满足条件的情况下，所有方案中保留的绳索强度之和的最小值。

输入格式

第一行两个数 n, m ，表示诡雷的数量以及增加绳索的次数。

接下来 m 行，第 i 行包含四个数 x_i, y_i, l_i, w_i ，表示第 i 次增加绳索时选定的参数。

输出格式

一个数，表示所有保留绳索的方案中绳索强度之和的最小值。

样例输入

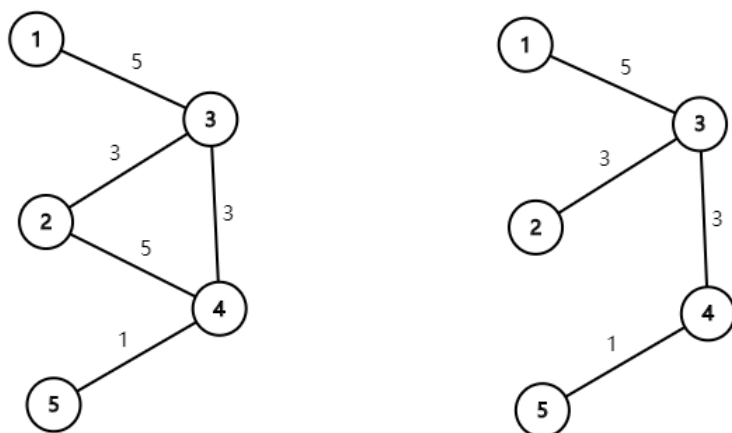
```
5 3
1 3 2 5
2 3 2 3
4 5 1 1
```

样例输出

```
12
```

样例解释

下图左部为诡雷之间连接绳索的情况，右部为一种最优的保留绳索的方案：



可以验证不存在更优的保留方案，因此绳索强度之和的最小值为 12。

数据规模及约定

对于前 10% 的数据，保证 $n \leq 10, m \leq 20$ 。

对于前 30% 的数据，保证 $n \leq 10^3, m \leq 2 \times 10^3$ 。

对于另外 10% 的数据，保证 $l_i = 1$ 。

对于另外 25% 的数据，保证 $x_1 = 1, y_1 = 2, l_1 = n - 1, w_1 = 1$ ，并且 $w_i = 0$ 对 $i > 1$ 成立。

对于 100% 的数据，保证 $1 \leq n \leq 10^5, 1 \leq m \leq 5 \times 10^5, x_i \neq y_i, \max(x, y) + l - 1 \leq n, 0 \leq w_i \leq 10^9$