A. 最小公倍树

时间限制： 1.0 秒

空间限制： 512 MiB

相关文件： 题目目录

题目背景

听说有人嫌题面描述都太长了。

题目描述

对于任意 $V \subset \mathbb{N}^*$，$|V| < +\infty$，构造一张无向完全图 $G = (V, E)$，其中 (u, v) 的边权为 u, v 的最小公倍数 $\operatorname{lcm}(u, v)$。称 G 的最小生成树为 V 的最小公倍树（LCT, Lowest Common Tree）。

现在给出 L, R，请你求出 $V = \{L, L+1, \cdots, R\}$ 的最小公倍树 $LCT(V)$。

输入格式

从标准输入读入数据。

输入仅一行，包括两个正整数 L, R。

输出格式

输出到标准输出。

输出一个正整数，表示 $LCT(V)$ 的边权和。

样例1输入

```
3 12
```

样例1输出

```
126
```

样例1解释

其中一种最小公倍树上的边为 $(3,4),(3,5),(3,6),(3,7),(4,8),(3,9),(5,10),(3,11),(3,12)$。

样例2输入

```
6022 14076
```

样例2输出

```
66140507445
```

样例3输入

```
13063 77883
```

样例3输出

```
3692727018161
```

样例4输入

```
325735 425533
```

样例4输出

```
1483175252352926
```

子任务

对于 100% 的数据，保证 $1 \le L \le R \le 10^6$，且 $R - L \le 10^5$。

B. 骰子旅行

时间限制： 1.0 秒

空间限制： 512 MiB

相关文件： 题目目录

题目描述

在乐队 f 开巡演之前，按照惯例是要先组织乐队成员进行骰子旅行放松身心的。一次骰子旅行包括 N 个地点，这些地点分别标号为 $1, 2, \cdots, N$。乐队成员们事先约好在 s_0 处集合；而到了骰子旅行当天，大家都来到了集合地点 s_0，骰子旅行就算正式开始了。

骰子旅行的一大乐趣就是由骰子决定旅行的下一个目的地。当然，这个骰子不一定非得是六面的。我们可以认为，如果当前乐队成员们位于地点 i，那么下一个目的地会等概率地从 m_i 个互不相同的候选地点中产生，这些候选地点分别是 $l_{i,1}, l_{i,2}, \cdots, l_{i,m_i}$。我们记第 t 次投掷的结果是 s_t，那么第 $(t+1)$ 次将会前往 s_t 处掷骰子。第 1 次投掷在起点 s_0 处进行；而由于乐队之后还需要为了巡演排练，事先约定无论前往了哪些地点，投掷完第 T 次骰子，前往 s_T 后骰子旅行都得结束。

当然，享受 $s_0, s_1, \cdots, s_T$ 这些景点也是骰子旅行的一大乐趣。无论是否之前来过，每次到一个地点 s_t，乐队成员们都会尽情地浏览美景，品尝美食。只是如果之前来过 s_t，负责掷骰子的键盘手 S 在掷这第 $(t+1)$ 次骰子之前一定会说：“上次来到 s_t 仿佛还是上一次 t'，上一次在这里掷出了 $s_{t'+1}$，不知道这一次会掷出什么结果。”鼓手 Y 特别喜欢废话梗，所以每次 S 说这句话时，他都会把 $s_{t'+1}$ 记下来。特别地，如果 s_T 是之前经过的地点，那么 S 会说：“上次来到 s_t 仿佛还是上一次 t'，上一次在这里掷出了 $s_{t'+1}$，不过这一次就不投掷了，因为骰子旅行到这里就要告一段落了。”当然，Y 也会把这个 $s_{t'+1}$ 记下来。

作为这次骰子旅行的总结，Y 会把所有记录下来的 $s_{t'+1}$ 加起来，作为 S 的废话指数。

f 的下一次巡演马上就要开始了，于是 S 又盘算着带大家去参加骰子旅行。听说你是 f 的粉丝，S 找到了你，希望你能帮他算一下他这次骰子旅行的废话指数的期望值。

输入格式

从标准输入读入数据。

输入的第一行包括三个正整数 N, s_0, T，分别表示可能涉及地点的个数，骰子旅行的起点和骰子旅行中投掷骰子的次数。

接下来 N 行，第 i $(1 \le i \le N)$ 行先输入一个正整数 m_i，表示地点 i 的下一个目的地的候选地点数；接着输入 m_i 个正整数，表示这 m_i 个地点 $l_{i,1}, l_{i,2}, \cdots, l_{i,m_i}$。保证对于任意 i，m_i 个输入的地点互不相同。

输出格式

输出到标准输出。

输出一个数表示废话指数的期望。假设废话指数的期望化为最简分式后的形式为 p/q（即其中 p, q 互质），请输出 x 使得 $qx \equiv p \pmod{998,244,353}$ 且 $0 \le x < 998,244,353$。可以证明，在本题数据范围下，x 存在且唯一。

样例1输入

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

样例1输出

```
499122178
```

样例1解释

对答案有贡献的方案为：从点 1 出发走到 $2, 3, 4$ 中的任意一个点并返回点 1。对于某个点 $i(i = 2, 3, 4)$，走到点 i 并返回点 1 的概率为 $1/6$，而贡献为 i，故期望为 $\frac{1}{6} \times (2 + 3 + 4) = \frac{3}{2}$.

由 $499122178 \times 2 = 998244356 \equiv 3 \pmod{998244353}$ 可知 $3/2$ 在模 $998, 244, 353$ 意义下为 $499, 122, 178$，所以正确输出为 $499, 122, 178$。

样例2输入

```
7 1 4
6 2 3 4 5 6 7
6 1 3 4 5 6 7
6 1 2 4 5 6 7
6 1 2 3 5 6 7
6 1 2 3 4 6 7
6 1 2 3 4 5 7
6 1 2 3 4 5 6
```

样例2输出

```
274979351
```

样例2解释

转换前的答案为 $1625/432 \approx 3.761574$，而 $432 \times 274979351 = 118791079632 \equiv 1625 \pmod{998244353}$，所以模意义下的答案为 274979351。

样例3

见题目目录下的 *3.in* 与 *3.ans*。

子任务

对于 100% 的数据，保证 $1 \le N \le 100$，$1 \le T \le 100$，$1 \le s_0 \le N$，$1 \le m_i \le N$，$\sum_{i=1}^{N} m_i \le 5000$，$1 \le l_{i,j} \le N$，且 $\forall 1 \le i \le N, \forall 1 \le j_1 < j_2 \le m_i, l_{i,j_1} \ne l_{i,j_2}$。
$\sum_{i=1}^{N} m_i \le 1200$，$1 \le l_{i,j} \le N$，且 $\forall 1 \le i \le N, \forall 1 \le j_1 < j_2 \le m_i, l_{i,j_1} \ne l_{i,j_2}$。

C. 赛程制定

时间限制： 10.0 秒

空间限制： 512 MiB

相关文件： 题目目录

题目描述

Lewis热爱打拳，因此他组建了一家拳击俱乐部，希望通过举办表演赛卖票以筹集自己训练的资金，但是，很遗憾，Lewis人缘并不好，因此这个俱乐部只有两个成员——Lewis和他的好基友Valtteri，而观众们很快就厌倦了每晚都是他们两人出场的表演赛，票卖不出去了，拳击俱乐部濒临倒闭。穷则思变，Lewis决定通过请外援的方式，尝试拯救自己的俱乐部。

通过支票攻势，Lewis很快就请到了两个拳坛明星——Max和Checo，他们将作为飞行嘉宾加入Lewis的俱乐部。在接下来的一个赛季里，Lewis总共会安排$n\ (1 \le n \le 2*10^5)$ 场比赛，每场比赛会从现有的4个成员中选出2人比赛，对于第$i\ (1 \le i \le n)$场比赛，如果是Lewis和Valtteri的比赛，只能卖出a_i元的门票，如果是他们中一人和一个明星的比赛，能卖出b_i元的门票，如果是两个明星Max和Checo之间的比赛，能卖出c_i元的门票。观众们喜欢看明星之间的高水平比赛，而不是Lewis和Valtteri的菜鸡互啄，因此有$1 \le a_i < b_i < c_i \le 10^9$ 。除此之外，安排比赛时还有如下要求——

1. 因为明星都是日理万机的，他们只同意分别在Lewis的俱乐部停留最多t_m, t_c场比赛的时间。设Max出席的第一场比赛是第p_m场，最后一场是q_m场，Checo出席的第一场比赛是第p_c场，最后一场是q_c场，则需要满足$q_m - p_m + 1 \le t_m$ 且 $q_c - p_c + 1 \le t_c$；
2. Lewis深知自己不会是两个明星的对手，他不会愿意自己被打得鼻青脸肿直接KO，因此，他不会安排自己与两位明星的比赛（言下之意，挨打的工作就被Lewis偷偷安排给了自己的好基友Valterri，让我们为可怜的工具人Valterri默哀）；

同时，Lewis希望最大化自己的总收入，但是，他不太聪明，因此，如果一种方案满足以下条件，他就认为此方案满足”收入最大“——

定义 (“Lewis的最优方案”)：对于一种方案，可以看作一个长度为$2n$的序列，序列的第$(2i-1)$和$2i$项为第i场比赛的对阵双方，如果通过修改序列的任意**1**个位置，都无法得到一个收入严格大于当前收入的合法方案，则称此方案为“Lewis的最优方案”。

聪明的你很快就发现了，“Lewis的最优方案”不一定能最大化总收入且可能不唯一。已知Lewis会从所有“Lewis的最优方案”（两个方案相同，当且仅当每一天的对阵均相同，注意，Max vs Valtteri 和 Checo vs Valtteri虽然卖出门票相当，但视为两种不同的方案）等概率随机选一个方案执行，那么，请问在Lewis可能最终选择的所有方案中，门票收入的中位数是多少呢？（答案保留一位小数输出）

输入格式

从标准输入读入数据。

第 1 行：3个正整数n, t_m, t_c，意义见题面；

接下来 n 行：每行3个正整数 a_i, b_i, c_i 表示三种情况下卖出的门票价格。

输出格式

输出到标准输出。

一行一个正整数表示在Lewis最终选择的所有方案中，门票收入的中位数。

样例1输入

```
2 1 1
1 10 100
1 2 3
```

样例1输出

```
12.0
```

样例1解释

Lewis的“收入最大方案”总共有以下4种：

1. Max vs Checo, Lewis vs Valtteri，门票收入为100+1=101
2. Valtteri vs Max, Valtteri vs Checo，门票收入为10+2=12
3. Valtteri vs Checo, Valtteri vs Max，门票收入为10+2=12
4. Lewis vs Valterri, Max vs Checo，门票收入为1+3=4

中位数为(12+12)/2 = 12

样例2输入

```
3 1 3
1 2 3
5 6 12
1 5 6
```

样例2输出

```
14.0
```

子任务

数据范围：$1 \leq n, t_m, t_c \leq 2 * 10^5$，$1 \leq a_i < b_i < c_i \leq 10^9$

D. 造计算机

时间限制： 1.0 秒

空间限制： 512 MiB

相关文件： 题目目录

题目描述

小R和小C听说贵系有一门造计算机的课之后吓得连夜提交了退学申请。

开玩笑的啦！正处于大一的他们对这门课不但不害怕，甚至有些想笑。他们超强的动手能力甚至驱使他们想造一个玩意玩玩。

当然由于他们毕竟才大一，计算机专业课基本上都没上过，经过长时间的艰苦奋战，他们终于造出了一个奇怪的玩意：

这台计算机只有 n 个内存单元，反而有足够多个寄存器。内存单元的编号从 1 到 n ，寄存器从 $n+1$ 开始往上编号。每个内存单元和寄存器可以存储一个整数。

目前他们已经设计好了一类指令：`swap i, j`，表示交换编号为 i 和 j 的单元里的数，其中 i 和 j 均为正整数且 $i \neq j$ 。他们打算写一段程序来测试这条指令。

最开始， n 个内存单元中乱序存放着 $1 \sim n$ 这些数，且每个数恰好出现一次。而每个寄存器里存放的是它的编号。

两人打算设计一段指令序列，使得计算机依次执行完这些指令后，所有内存和寄存器中的数都归位，也就是恰好等于它自己的编号。

虽然没学过计算机专业课，小R和小C还是懂一点皮毛的，因此他们规定每条`swap`指令操作的两个位置至少有一个需要是寄存器，也就是 i 和 j 至少有一者应当大于 n。

然而，正当他们写完程序开始运行时，却发现系统崩溃了！在查找了半天原因后，他们发现了一个奇怪的bug：他们设计出来的计算机不能运行两条相同的指令！也就是说，他们不能在一段程序里出现两条相同的`swap i, j`指令。更进一步他们发现即使出现一条`swap i, j`一条`swap j, i`也不行，因为计算机会自动将这两条指令视为同一条。

然后可怜的小R和小C就斯巴达了。不过他们在弃疗之前还是打算利用现有的架构把程序写出来。不仅如此，他们还希望用到的寄存器数量尽可能少。你能帮帮他们吗？

输入格式

从标准输入读入数据。

第一行：一个正整数 n $(1 \leq n \leq 10^5)$ 表示内存单元的数量。

第二行： n 个正整数a_i 依次表示第 i 个内存单元中的初始值，保证所有 a_i 构成一个 $1 \sim n$ 的排列。

输出格式

输出到标准输出。

第一行：两个非负整数m, k，分别表示你用到的寄存器数量和指令的条数。

接下来k行，每行输出两个正整数i, j，依次表示你设计的每一条指令中交换的两个位置。你需要保证 $1 \leq i, j \leq n+m$ ，并且满足题目中对于指令的限制条件。

如果有多种设计指令的方案满足题目所需，输出任意一种即可。

你需要最小化 m 而无需最小化 k ，但需要保证$k \leq 10^6$。输入数据保证符合要求的解存在。

样例1输入

```
2
2 1
```

样例1输出

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

样例1解释

最初，前 4 个单元的值依次为 $(2,1,3,4)$ 。

执行指令`swap 3, 4`，各单元的值变为 $(2,1,4,3)$ 。

执行指令`swap 1, 3`，各单元的值变为 $(4,1,2,3)$ 。

执行指令`swap 2, 4`，各单元的值变为 $(4,3,2,1)$ 。

执行指令`swap 1, 4`，各单元的值变为 $(1,3,2,4)$ 。

执行指令`swap 2, 3`，各单元的值变为 $(1,2,3,4)$ 。

可以证明 $m=1$ 是不行的。

E. 搬砖

时间限制： 10.0 秒

空间限制： 512 MiB

相关文件： 题目目录

题目背景

张华考上了北京大学；李萍进了中等技术学校；小E在工地搬砖：他们都有光明的前途。

题目描述

温馨提示：请不要模仿小E的搬砖方式，那样很累。

为了能够快乐地搬砖，小E有一种特殊的搬砖方式。

假设他的面前有 n 摞砖，他会在一个小时内搬走每一摞砖最上面的 d 块。其中 d 是小E当前的精力值。如果一摞砖不够 d 块，小E会把这一摞砖剩下的所有砖搬走。

当小E工作完一个小时后发现自己搬完了至少一摞砖，那么他会觉得很快乐，并且继续工作一个小时；但是由于完成了一部分工作，小E可能会产生懈怠的心理，导致精力值有所下降。具体地，对于每一摞砖都有一个属性 b，当小E搬完这一摞砖后，精力值就会下降 b。

如果没有任何一摞砖被搬完，小E就会停止工作。如果精力值下降到 0 或以下，小E也会停止工作。如果小E发现自己需要工作但是所有的砖已被搬完，他会用别的方式来度过这一小时，但这一小时仍算作小E的工作时间。

工地的砖在不停增加，问如果小E初始的精力值为 d，那么他可以连续工作几个小时？

输入格式

第一行一个正整数 T，表示事件总数。

接下来 T 行，每行若干个整数，其中第一个整数为 op 表示事件类型。

若 $op = 1$，则后面跟着两个整数 a, b，表示新增了一摞砖，砖有 a 块，搬完后小E的精力值会下降 b。

若 $op = 2$，则后面跟着一个正整数 d，询问若小E初始的精力值为 d，那么他可以连续工作几个小时。

注意 $op = 2$ 的操作**不会**改变任何一摞砖的数量。

输出格式

对于每个询问，输出一行一个整数表示答案。

样例1输入

```
5
1 6 1
1 3 0
1 9 2
2 3
2 4
```

样例1输出

```
3
4
```

样例1解释

第一组询问：

初始有 3 摞砖，数量分别为 $(6,3,9)$，小E的初始精力是 3。

第一个小时，小E在每一摞砖中各搬了 3 块，数量变成 $(3,0,6)$。其中第二摞砖被搬完，小E的精力因此下降 0 并且继续工作一个小时。

第二个小时，小E在每一摞砖中各搬了 3 块，数量变成 $(0,0,3)$。其中第一摞砖被搬完，小E的精力因此下降 1 并且继续工作一个小时。

第三个小时，小E在每一摞砖中各搬了 2 块，数量变成 $(0,0,1)$。由于没有新的砖摞被搬完，小E停止工作。

第二组询问：

初始有 3 摞砖，数量分别为 $(6,3,9)$，小E的初始精力是 4。

第一个小时，小E在每一摞砖中各搬了 4 块（第二摞砖由于只有 3 块就只搬了 3 块，以下省略），数量变成 $(2,0,5)$。其中第二摞砖被搬完，小E的精力因此下降 0 并且继续工作一个小时。

第二个小时，小E在每一摞砖中各搬了 4 块，数量变成 $(0,0,1)$。其中第一摞砖被搬完，小E的精力因此下降 1 并且继续工作一个小时。

第三个小时，小E在每一摞砖中各搬了 3 块，数量变成 $(0,0,0)$。其中第三摞砖被搬完，小E的精力因此下降 2 并且继续工作一个小时。

第四个小时，小E在每一摞砖中各搬了 1 块，但其实此时已经没有砖了，不过这一小时仍然算进小E的工作时间。由于没有新的砖摞被搬完，小E停止工作。

样例2输入

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

样例2输出

```
2
1
```

样例2解释

第一组询问：

初始有 1 摞砖，数量为 2，小E的初始精力是 2。

第一个小时，小E在每一摞砖中各搬了 2 块，数量变成 0。这一摞砖被搬完，小E的精力因此下降 1 并且继续工作一个小时。

第二个小时，小E在每一摞砖中各搬了 1 块，但其实此时已经没有砖了，不过这一小时仍然算进小E的工作时间。由于没有新的砖摞被搬完，小E停止工作。

第二组询问：

初始有 2 摞砖，数量为 $(2,2)$，小E的初始精力是 2。

第一个小时，小E在每一摞砖中各搬了 2 块， 数量变成 $(0,0)$。两摞砖都被搬完，小E的精力因此下降 $1+1=2$。由于小E的精力下降到 0，他停止工作。

子任务

保证 $T \le 351493, 1 \le op \le 2, 1 \le a \le 100000, 0 \le b \le 100000, 1 \le d \le 100000$。

F. 喵喵花园

时间限制： 5.0 秒

空间限制： 512 MiB

相关文件： 题目目录

题目背景

题目描述

喵喵是一只非常富有的猫咪，他在海淀区拥有一个大花园。

这个大花园是由一些旧栅栏为边界所形成的 N-gon（即具有 N 边的多边形）。

由于圣诞节快到了，喵喵想用K棵圣诞树来装饰一下花园。
同时，喵喵坚信找到一些好的位置来种树会给他带来好运。

作为一只好猫咪，他决定寻找最佳位置如下：

- 所有的树都应该在花园的边界上。
- 这些 K 树应该平均划分花园的周长。
- 由树木形成的新凸面K-gon 的面积应尽可能小。

虽然喵喵比你有钱，但他没有你那么聪明。
因此，他给了你一些钱，让你帮他找出凸K-gon 的最小面积。

输入格式

第一行包含两个整数，N 和 K，代表原本花园边界的顶点数和树的数量。

接下来的每行 N 行包含两个整数 x_i 和 y_i，表示花园边界顶点的坐标。

所有座标均为逆时针给出的。

输出格式

输出凸面K-gon 的最小面积。
如果相对或绝对误差不超过 10^{-8}，则您的答案被认为是正确的。

样例1输入

```
5 4
0 0
1 0
2 1
2 2
0 2
```

样例1输出

```
1.9892766953
```

样例1解释

样例2输入

```
3 3
0 0
0 1
1 0
```

样例2输出

```
0.1226170434
```

样例2解释

子任务

- $3 \le N, K \le 1000$

- $-10^5 \le x_i, y_i \le 10^5$

G. 挑战

时间限制： 1.0 秒

空间限制： 512 MiB

相关文件： 题目目录

题目描述

足够聪明的 Alice 和 Bob 在玩一种棋盘游戏。这个游戏需要用到一个有 $(n+1)$ 个格子的长条棋盘，按从左到右的顺序给每个格子编号 $0, 1, \cdots, n$。除了第 n 格以外，每一格都有两个数 p_i, q_i。游戏开始前，将一个棋子放在第 0 格。游戏由二人轮流操作，这里我们不妨假设 Alice 先手。

轮到其中一位玩家进行操作时，这位玩家可以根据当前格子的 p 值决定前进的步数。具体地说，假设当前棋子位于第 k 格，那么当前进行操作的玩家可以将棋子向前移动 x 格，其中 x 可以是满足 $1 \le x \le p_k$ 的任意整数。如果玩家没有走满 p_k 格，即 $x < p_k$，那么该玩家可以在完成移动后选择是否进行一次挑战。如果选择不进行挑战，那么由另一位玩家进行下一轮操作。否则，如果当前玩家选择挑战，那么系统将会产生两个随机**整数** u 和 v，其中：u 表示挑战的能量，它在 $[1, p_k - x]$ 中等概率产生；v 表示挑战所需的活化能，它在 $[0, q_k + q_{k+x}]$ 中等概率产生。根据 u 和 v 的值，系统会根据以下规则自动判定挑战结果：

- 如果 $u > v$，则挑战成功，对方玩家的操作被跳过一轮，由当前玩家继续操作；
- 如果 $u = v$，则挑战结果为平手，什么事情都不会发生，由对方玩家进行操作；
- 如果 $u < v$，则挑战失败，当前玩家下一轮操作将会被跳过，即对方玩家可以连续操作两轮。

为了防止其中一方玩家一直被跳过，规定：

- 如果当前玩家通过自身的挑战获得额外操作机会，则该玩家在该额外操作机会中不能进行第二次挑战；
- 如果当前玩家通过对方玩家的挑战获得额外操作机会，则该玩家不能在其第一次操作结束时发起挑战，只能在第二次操作结束时选择是否进行挑战，并且当且仅当挑战成功时可以进行第三次操作。

需要注意的是，无论连续进行多少次操作，每次操作都需要将棋子向前移动至少 1 格。同大多数游戏一样，谁将棋子移动到终点（即第 n 格）谁就获胜。

Alice 和 Bob 都足够聪明，可以心算出对于当前棋子的位置，能使自己获胜概率最大的操作。作为一名旁观者，你没有他们那么强的心算能力；但是你也想通过自己编程的能力，计算出当 Alice 先手从第 0 格开始进行操作时，Alice 的胜率。

输入格式

从标准输入读入数据。

输入的第一行包含一个正整数 n，表示棋盘包含 $(n+1)$ 格，分别标号 $0, 1, \cdots, n$。

输入的第二行包含 n 个正整数 $p_0, p_1, \cdots, p_{n-1}$，分别表示第 0 格至第 $(n-1)$ 格的 p 值。

输入的第三行包含 n 个正整数 $q_0, q_1, \cdots, q_{n-1}$，分别表示第 0 格至第 $(n-1)$ 格的 q 值。

输出格式

输出到标准输出。

输出一个实数，表示 Alice 先手开始游戏的胜率。当你的输出与标准输出的绝对误差不超过 10^{-6} 时，我们认为你的输出是正确的。

样例1输入

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

样例1输出

```
1.000000000000000000
```

样例1解释

Alice 先手，由于可以直接从第 0 格移动到终点的第 3 格，Alice 会直接将棋子移动到第 3 格，故 Alice 必胜。

样例2输入

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

样例2输出

```
0.250000000000000000
```

样例2解释

Alice 先手，但是不能直接移动到第 3 格，并且无论结束操作时棋子在第 1 格还是第 2 格，Bob 都可以直接将其移动到终点的第 3 格，因此 Alice 必须尝试挑战。将棋子移动到第 1 格并发动挑战，挑战成功的概率为 $1/4$，故 Alice 的胜率为 $1/4$。

样例3输入

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

样例3输出

```
0.300000000000000000
```

子任务

对于 100% 的数据，保证 $1 \le n \le 100,000$，$1 \le p_i, q_i \le 333$。

H. tdnmo

时间限制： 1.0 秒

空间限制： 512 MiB

相关文件： 题目目录

题目描述

给定一棵 n 个顶点的有根树，顶点编号为 $1, 2, \ldots, n$，1 号顶点为根。定义有向邻域 $N(x, y)$ 为以 x 为根的子树中，距离 x 小于 y 的顶点构成的集合，其中 $1 \le x \le n$，$0 \le y \le n$，x, y 为整数。

给出 m 个有向邻域 $N(x_i, y_i)_{i=1}^m$，你可以从 $N(1, 0)$ 出发，经过不超过 $5m$ 次操作到达每个给出的有向邻域，可以使用的操作有：

1. 从有向邻域 $N(x, y)$ 移动到 $N(x', y')$，满足 $N(x, y) \subseteq N(x', y')$；
2. 撤销一次1操作，即回到之前最后一次未撤销的1操作前的位置，并将这次1操作标为已撤销；
3. 声明当前到达了有向邻域 $N(x_i, y_i)$，满足当前所在邻域是 $N(x_i, y_i)$；

其中操作1的代价为移动前后两个有向邻域的元素个数之差，操作2和3不计代价，要求操作2执行时必须存在未撤销的操作1，操作3必须对每个 $1 \le i \le m$ 恰好各执行一次。

你需要保证操作的总代价不超过 2.5×10^9。

输入格式

从标准输入读入数据。

第一行两个整数 $n\ m$;

接下来一行，共 $n-1$ 个整数，依次表示编号为 $2,3,\ldots,n$ 的顶点的父亲 $f_2,\ldots,f_n$，保证父亲的编号小于孩子的编号;

接下来的 m 行中，第 i 行两个整数 $x_i\ y_i$，表示给出的每个有向邻域。

输出格式

输出到标准输出。

第一行一个整数 m'，表示你进行的操作次数;

接下来 m' 行，依次表示每个操作;

操作1表示为 $1\ x'\ y'$;

操作2表示为 2;

操作3表示为 $3\ i$。

样例输入

```
8 4
1 1 1 2 2 2 5
2 1
1 1
6 0
1 2
```

样例输出

```
16
1 2 1
3 1
2
1 6 0
3 3
2
1 1 1
1 1 1
3 2
2
1 1 2
1 1 2
3 4
2
2
```

```
2
```

子任务

对于 100% 的数据，满足 $1 \le n, m \le 10^6$，$1 \le f_i \le i-1$，$1 \le x_i \le n$，$0 \le y_i \le n$，所有数值为整数。

I. 分组作业

时间限制： 1.0 秒

空间限制： 512 MiB

相关文件： 题目目录

题目描述

老师布置了分组作业。在此之前，老师将班上 $2n$ 个学生分成了 n 组，每组两个人。其中 1 号和 2 号为一组，3 号和 4 号为一组，……，$2n-1$ 号和 $2n$ 号为一组。

老师让每个队伍自行安排分工。这样是否合作就成了一个大问题。大家决定用表决的方式来确定。首先每个人决定是否愿意和队友合作。不同的人因为自己的原因和分配的队友的原因，对合作的意愿不一样，对于第 i 个学生，选择“愿意”会产生 c_i 的不满，选择“不愿意”会产生 d_i 的不满。

如果两名队友都选择“愿意”，那么根据实际情况他们可以合作或者不合作。但是如果有一名队友选择“不愿意”，那么他们只能不合作。

学生中还有 m 个单向的喜欢关系，一个关系形如“A 喜欢 B”。在这样一个关系中，如果 A 没有和队友合作，且 B 选择了“愿意”，A 会有略微沮丧，产生 a_i 的不满；如果 A 表决了“不愿意”，但 B 成功与队友合作，那么 A 会羡慕嫉妒恨并产生 b_i 的不满。（由于当 A 和 B 在同一组时这种设定会变得很奇怪，所以题目保证不会有这种情况）其中 i 表示第 i 个关系。

如果一个学生 i 选择了“愿意”但是他的队友选择了“不愿意”，那么他会因为队友产生 e_i 的不满。

问所有情况下最小的不满之和是多少。

输入格式

从标准输入读入数据。

第一行两个整数 n, m。

接下来 $2n$ 行，每行三个整数 c_i, d_i, e_i。

接下来 m 行，每行四个正整数 A, B, a_i, b_i 。

输出格式

输出到标准输出。

一行一个整数表示答案。

样例输入

```
2 1
8 6 7
5 2 8
7 1 5
6 5 8
1 4 4 3
```

样例输出

```
14
```

子任务

保证 $1 \le n \le 5000$，$0 \le m \le 10000$，$1 \le a_i, b_i, c_i, d_i, e_i \le 10^9$。

J. THUPC

时间限制： 1.0 秒

空间限制： 512 MiB

相关文件： 题目目录

题目描述

一年两度的THUPC又要来了，小C和小Z作为参赛无数届的老选手，自然也要来凑一番热闹。不过因为已经是老年人了，参赛自然是免谈了，但是他们对THUPC几年不换的logo产生了审美疲劳，为了更好地吸引大家报名，他们打算重画一个更花里胡哨的。

笑死，你怎么可能指望两个宅男码农有丝毫的艺术细菌?

他们深知这件事不在自己的能力范围之内，于是打算写一个人工智障来帮自己来画logo！

经过不懈的努力，他们的人工智障终于能跑起来了，不过他们很快就发现这个人工智障比自己还没有艺术细菌——它只会在平面上画水平和竖直的线段来拼成“THUPC”字样！

不过程序写都写了，不用白不用。小C和小Z针对程序的这一特性进行深入研究后制定了这样的规则：

对于每一条水平线段，设其横坐标区间为 $[l_i, r_i]$ ，纵坐标为 y_i； 对于每一条竖直线段，设其纵坐标区间为 $[d_i, u_i]$ ，横坐标为 x_i 。上述所有数值均为整数，且满足$r_i > l_i, u_i > d_i$。

“THUPC”字样应当由 15 条线段拼成，设其编号为 $1 \sim 15$ 。对于每一个字母，规则如下：

字母"T"由 1 号水平线段和 2 号竖直线段组成，满足 $d_2 < y_1 = u_2, l_1 < x_2 < r_1$ 。

字母"H"由 3 号竖直线段、4 号水平线段和 5 号竖直线段组成，满足 $d_3 = d_5 < y_4 < u_3 = u_5, x_3 = l_4 < r_4 = x_5$ 。

字母"U"由 6 号竖直线段、7 号水平线段和 8 号竖直线段组成，满足 $d_6 = d_8 = y_7 < u_6 = u_8, x_6 = l_7 < r_7 = x_8$ 。

字母"P"由 9 号竖直线段、10 号水平线段、11 号水平线段和 12 号竖直线段组成，满足 $d_9 < y_{11} = d_{12} < u_9 = y_{10} = u_{12}, x_9 = l_{10} = l_{11} < r_{10} = r_{11} = x_{12}$ 。

字母"C"由 13 号竖直线段、14 号水平线段和 15 号水平线段组成，满足 $d_{13} = y_{15} < u_{13} = y_{14}, x_{13} = l_{14} = l_{15} < r_{14} = r_{15}$ 。

生成的这 5 个字母可以排布在平面的任何地方而无需从左到右排列，但是组成任意两个不同字母的任意两条线段不得相交。

需要注意的是，人工智障给出的线段顺序可能并不按照上述编号顺序；另外，给出的线段可能出现同方向线段的首尾相连、重叠或包含，此时应将其视为连续的一整条线段。

只有生成的线段在连接和排序后符合上述规范，才认为人工智障生成了一幅正确的logo；否则，如果出现多余的线段、缺少某条线段或坐标不满足要求等情况均为不正确的。

最后，小C和小Z要写一个程序来检验人工智障的每一份输出结果是否符合上述规范，不过熬夜连肝三天的他们终于累得爬不起来了，于是他们请你来帮忙。

输入格式

从标准输入读入数据。

第1行：一个正整数 n 表示线段的个数，保证 $1 \le n \le 10^5$ 。

接下来 n 行，每行先输入一个整数 op_i ，必定为 0 或 1 ：如果 $op_i = 0$ ，表示第 i 条线段为水平线段，接下来输入 3 个整数 l_i, r_i, y_i 描述这条线段，保证 $l_i < r_i$ ；如果 $op_i = 1$ ，表示第 i 条线段为竖直线段，接下来输入 3 个整数 d_i, u_i, x_i 描述这条线段，保证 $d_i < u_i$ 。

保证输入的坐标均在 $[-10^9, 10^9]$ 范围内。

输出格式

输出到标准输出。

如果符合规范，输出一个字符串 `Yes`，否则输出一个字符串`No`。

样例1输入

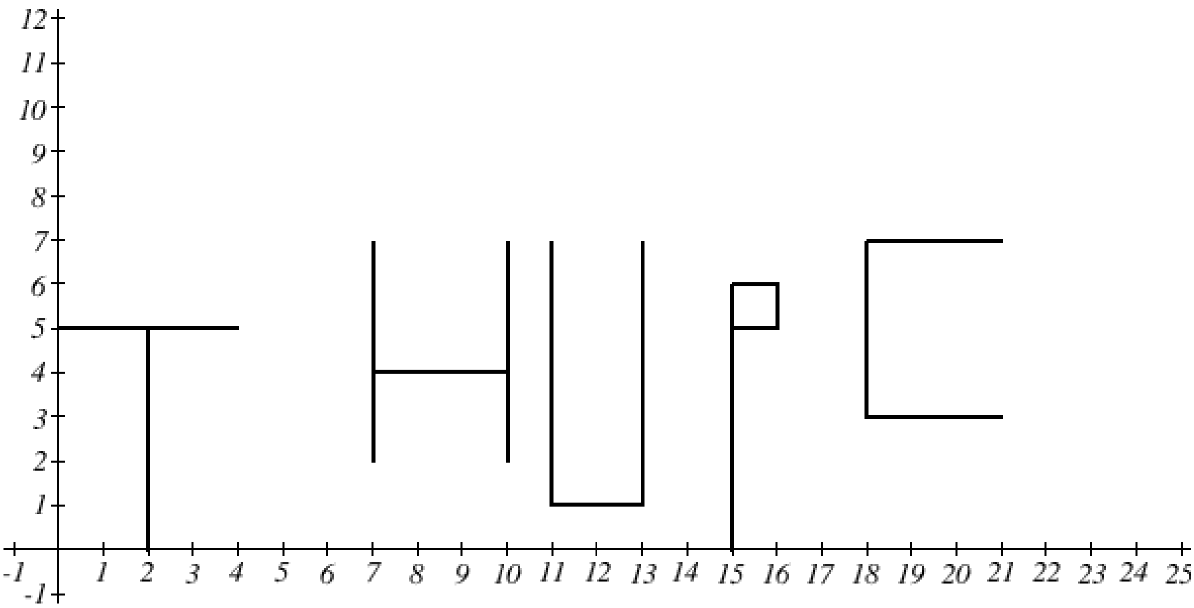
```
17
1 0 5 2
0 0 3 5
0 3 4 5
1 2 7 7
1 2 7 10
0 7 10 4
0 11 13 1
```

```
1 1 7 11
1 1 7 13
1 0 6 15
0 15 16 5
0 15 16 6
1 5 6 16
1 3 6 18
1 4 7 18
0 18 21 3
0 18 21 7
```

样例1输出

```
Yes
```

样例1解释



这组样例中，字母“T”的水平线段和“C”的竖直线段分别是由两条线段拼成的。

K. 数正方体

时间限制： 1.0 秒

空间限制： 512 MiB

相关文件： 题目目录

题目背景

学~霸~题

数正方体

题目描述

小E有一块面积为 $n \times m$ 的矩形区域，上面有 $n \times m$ 个边长为 1 的格子，第 i 行 j 列的格子上堆了 A_{ij} 个同样大小的正方体积木。小E做了某道题之后，突发奇想把这些正方体画成了字符画，然后让你帮他数一数他一共有多少个正方体。我们定义每个正方体为如下格式，并且不会做任何旋转，只会严格以这一种形式摆放：

```
..+---+
./   /| 高
+---+ |
|   | +
|   |/.宽
+---+..
 长
```

每个顶点用 1 个 `+` 表示，长用 3 个 `-` 表示，宽用 1 个 `/` 表示，高用两个 `|` 表示。字符 `.` 作为背景。中间的空白是空格（ASCII码为 32 ）。

若两个正方体左右相邻，图示为

```
..+---+---+
./   /   /|
+---+---+ |
|   |   | +
|   |   |/.
+---+---+..
```

若两个正方体积木上下相邻，图示为

```
..+---+
./   /|
+---+ |
|   | +
|   |/|
+---+ |
|   | +
|   |/.
+---+..
```

若两个正方体前后相邻，图示为

```
....+---+
.../   /|
..+---+ |
./   /| +
+---+ |/.
|   | +..
|   |/...
+---+....
```

位于前面的正方体的面会遮挡住位于后面的正方体的面。为了让你看得清楚，没有整列正方体被挡在后面，小E保证了 $1 \le A_{ij} \le A_{i-1,j}$，$1 \le A_{ij} \le A_{i,j-1}$。并且图中没有整行或者整列的 `.` 。所以，一个字符画对应唯一的矩阵 A，一个矩阵 A 也对应一个唯一的字符画。

输入格式

从标准输入读入数据。

第一行两个正整数 r, c，表示图的高度和宽度。（注意不是 n 和 m）

接下来是一个 r 行 c 列的字符画，表示小E堆叠的正方体。

输出格式

输出到标准输出。

一行一个整数，表示正方体的数量。

样例输入

```
14 17
....+---+---+....
.../   /   /|....
..+---+---+ |....
./   /|   | +---+
+---+ |   |/   /|
|   | +---+---+ |
|   |/   /|   | +
+---+---+ |   |/|
|   |   | +---+ |
|   |   |/   /| +
+---+---+---+ |/.
|   |   |   | +..
|   |   |   |/...
+---+---+---+....
```

样例输出

```
14
```

样例解释

此时 A 矩阵为 $\begin{bmatrix} 3 & 3 & 2 \\ 3 & 2 & 1 \end{bmatrix}$，因为 $3+3+3+2+2+1=14$，所以图中共有 14 个正方体。

子任务

保证 $1 \le n, m \le 50$， $1 \le A_{ij} \le 100$。（注意这里是 n 和 m 不是 r 和 c）

保证 $\forall 1 < i \le n, A_{ij} \le A_{i-1,j}$。

保证 $\forall 1 < j < m, A_{ij} \le A_{i,j-1}$。

保证字符画中没有一整行或者一整列是 `.` 。