

10.10NOIP模拟赛

题目名称	信号塔	vivo50	选举	修路
题目类型	全文比较	全文比较	自定义比较器	全文比较
输入文件名	beacon.in	game.in	t.in	road.in
输出文件名	beacon.out	game.out	t.out	road.out
测试点数量	20	10	1	10
每个测试点时限	1s	5s	2s	7s
内存限制	512M	1024M	1024M	1024M

编译选项

对于 C++ 程序 `-lm -O2 -std=c++14`

A. 信号塔（beacon）

题目描述

有 n 个村庄排成一排，依次标号为 $1, 2, \dots, n$ 。 i 号村庄的海拔为 h_i ， h 是一个 $[1, n]$ 的排列。

你现在希望将这些村庄划分成若干个非空连续段，并在每个连续段中海拔最高的村庄安装一个信号塔。村民们发现，当所有信号塔的海拔从左到右依次递增时，村庄之间的信息交流就会畅通无阻。

你需要求出有多少种合法的划分村庄的方式，使得所有信号塔的海拔从左到右依次递增。答案对 998244353 取模。

样例输入 1

```
3
2 3 1
```

样例输出 1

```
2
```

样例 1 解释

有两种合法的划分方式： $\{2\}, \{3, 1\}$ 和 $\{2, 3, 1\}$ 。

样例 2

见选手目录下的 `beacon/ex_beacon2.in` 与 `beacon/ex_beacon2.ans`。

该样例满足测试点 1 ~ 4 的限制。

样例 3

见选手目录下的 `beacon/ex_beacon3.in` 与 `beacon/ex_beacon3.ans`。

该样例满足测试点 13 ~ 20 的限制。

数据范围与提示

对于所有测试点： $1 \leq n \leq 5 \times 10^5$ ， $1 \leq h_i \leq n$ 且 h 是一个 $[1, n]$ 的排列。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	特殊性质
1 ~ 4	20	无
5 ~ 10	2000	无
11 ~ 12	5×10^5	A
13 ~ 20	5×10^5	无

特殊性质 A： $\forall i \in [1, n - 1], h_i < h_{i+1}$ 。

B. vivo50 (game)

题目描述

又是疯狂星期日，zhoutbbbb去吃cfk咯！

zhoutbbbb发现今天的cfk的血赚菜品构成了一棵树，每个菜品都有一个编号（所有菜品编号为 $1 \sim n$ 的一个排列）。

若 (u, v) 是树上的一条边，那么它的边权表示同时购买 u, v 后需要支付的钱。

今天cfk举办了节日活动，有特殊的购买规则：

- 选取两个菜品 u, v
- 购买树上 u, v 之间所有的菜品（包括 u, v ）

但zhoutbbbb比较挑食，他每次对 u 的选择总是在一个连续编号区间 $[p, q]$ 里。

现在他想问你，如果给定 v 的选取，以及 p, q ，那么他最少需要花多少钱？

由于zhoutbbbb带的钱不太多，因此保证所有 u, v 选取后，花费的米不会超过 10^9 米。

输入

第一行一个整数 $n(\leq 10^5)$ ，表示菜品个数。

之后 $n - 1$ 行，每行三个整数 u, v, w ，菜品 u, v 有一条直连边，同时购买它俩需要花费 w 米。

之后一行一个整数 q ，表示询问个数。

之后 $q(\leq 10^5)$ 行，每行三个整数 p, q, v ，表示询问。

输出

一共 q 行，每行一个整数表示对应询问的答案。

样例输入

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

样例输出

```
1
```

C. 选举 (t)

题目描述

Byteland 的选民们要进行总统选举。一共有 n 位选民和 m 位候选人，编号分别从 1 到 n 和从 1 到 m 。

每位选民有一个**非空**的喜欢的候选人的列表，这个列表是按照喜欢程度排序的。比如说一位选民的列表是 $\{2, 4, 7\}$ ，那么他最喜欢的候选人是 2 号候选人，次喜欢的是 4 号候选人，再次是 7 号候选人，其余候选人都不喜欢。

选举会进行若干轮，具体规则如下：

- 每一轮，每位选民会投给某位在自己列表中的候选人恰好一票。
- 第一轮，每位选民会投给自己最喜欢的候选人。
- 从第二轮开始，每位选民会投给**自己列表中上一轮**票数最多的候选人。如果有多位在列表中的候选人票数都是最多的，那么会投给**这些人中最喜欢的那一位**（即按照票数为第一关键字，喜欢程度为第二关键字排序）。
- 如果每位选民在第 i 轮 ($i > 1$) 投给的候选人都和在第 $i - 1$ 轮投给的候选人一样，那么选举会在第 i 轮结束后停止，并且 **random** 一位候选人作为总统。

聪明的你一定发现了这个选举没有什么蛋用，但是你只关心选举进行的轮数。

你需要构造出 n, m ，和每一位选民的喜欢的候选人的列表，使得最终选举进行的轮数尽可能多。

由于 Byteland 人数并不多，所以你还需要保证 $n, m \leq 1000$ 。由于 Byteland 不想让某位候选人得到 0 票，你还需要保证 $n \geq m$ ，但是不必要保证在实际选举中每位候选人的票数都 > 0 （也就是说这句话存在的意义仅仅就是要让 $n \geq m$ ）。你还需要保证所有人的喜欢的候选人的列表大小之和 ≤ 100000 。

输入格式

没什么输入。

输出格式

第一行两个整数 n, m 。

接下来 n 行，每行表示一位选民的表单：第一个数字 k 表示表单的大小，接下来 k 个数字表示他的表单，按照喜欢程度排序。

样例输出1

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

样例解释1

选举会进行 3 轮。

计分策略

本题仅有一个测试点。

如果输出不合法，得分为 0。

设 a 为选举进行的轮数， $S = \{5, 10, 20, 35, 50, 95, 145, 195, 240, 490\}$ ，设 x 为 S 中 $\leq a$ 的元素个数，那么得分为 $10x$ 。

时间及空间限制

2s, 1024MB

保证 $n = m = 1000$, $\sum k = 100000$ 时 `spj` 的耗时不超过 $0.3s$, 占用空间不超过 $24MB$ 。也就是说选手的可用时间限至少为 $1.7s$, 可用空间限制至少为 $1000MB$ 。

D. 修路(road)

题目描述

最近 X 国的交通状况非常糟糕, 所以 X 国打算重新修建道路。

X 国共有 n 个城市, 编号为 $1 \sim n$ 。

现在开发商给出了 m 条可供修建的道路, 第 i 条连接城市 u_i 和 v_i , 长度为 l_i 。

X 国由于不太想修建太多冗余的道路, 所以只打算修筑 $n - 1$ 条道路, 并且将所有城市连通起来。也就是说, 这 $n - 1$ 条道路要形成一棵生成树。

在修建道路之后, X 国需要规定一个 *中心城市* mid 。我们认为一个规划的 *拥挤指数* 为 $\sum_{i=1}^n dis(i, mid)$, 其中 $dis(u, v)$ 表示城市 u 到城市 v 的距离。

X 国认为还有一个麻烦: 有些城市会只被一条道路连接, 这使得这些城市的交通不方便, 会花费额外代价。

具体地, 如果最终的修路方案中有 k 个城市只被一条道路连接, 那么 *拥挤指数* 会增加 $k \times S$, 其中 S 为给定的常数。

现在 X 国首领想问你, 如何修建道路并规划 *中心城市* , 才能使 *拥挤指数* 最小呢? 同时他还想问你, 有几种不同的规划方案 (修路+规划 *中心城市*) 能使 *拥挤指数* 最小呢?

输入格式

第一行三个整数 n, m, S 。

接下来 m 行, 每行三个整数 u_i, v_i, l_i , 描述一条道路。

输出格式

一行两个整数 ans 和 cnt , 表示最小的 *拥挤指数* , 和达到最小 *拥挤指数* 的方案数。

样例输入1

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

样例输出1

```
11 1
```

样例解释1

只有一种规划方案最优: 选上 1, 3 号边, 并令 *中心城市* 为 2 号节点。

样例输入2

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

样例输出2

```
25 4
```

样例解释2

如果选择 1, 2, 5 号边，我们可以令 *中心城市* 为 1 号节点或者 2 号节点。

如果选择 1, 3, 4 号边，我们可以令 *中心城市* 为 1 号节点或者 2 号节点。

共 4 种方案。

样例输入3

```
2 1 1
1 2 1
```

样例输出3

```
3 2
```

样例解释3

用于查出某些错误。

数据范围

对于 30% 的数据，有 $n \leq 5$ 。

对于 50% 的数据，有 $n \leq 11$ 。

对于额外 10% 的数据，有 $m = \frac{n \times (n-1)}{2}$, $S = 0$, $l_i = 0$ 。

对于额外 10% 的数据，有 $m = \frac{n \times (n-1)}{2}$, $S = 0$, $l_i = 1$ 。

对于额外 10% 的数据，有 $m = \frac{n \times (n-1)}{2}$, $S = 10^9$, $l_i = 1$ 。

对于所有 100% 的数据，有 $1 \leq n \leq 15$, $n-1 \leq m \leq \frac{n \times (n-1)}{2}$, $0 \leq l_i$, $S \leq 10^9$

在一个测试点中，如果你**输出了两个整数**，但是只有第一个正确，那么可以获得 60% 的分数。

保证没有重边和自环。

保证将所有边都修建以后，所有城市连通。

