

A 奶

时空限制：1s 256MB

文件名

preference.in/preference.out/preference.cpp

题目描述

Farmer John 计划建造 N 个农场，用 $N - 1$ 条道路连接，构成一棵树（也就是说，所有农场之间都互相可以到达，并且没有环）。每个农场有一头奶牛，品种为更赛牛或荷斯坦牛之一。

Farmer John 的 M 个朋友经常前来拜访他。在朋友 i 拜访之时，Farmer John 会与他的朋友沿着从农场 A_i 到农场 B_i 之间的唯一路径行走（可能有 $A_i = B_i$ ）。除此之外，他们还可以品尝他们经过的路径上任意一头奶牛的牛奶。由于 Farmer John 的朋友们大多数也是农场主，他们对牛奶有着极强的偏好。他的有些朋友只喝更赛牛的牛奶，其余的只喝荷斯坦牛的牛奶。任何 Farmer John 的朋友只有在他们访问时能喝到他们偏好的牛奶才会高兴。

请求出每个朋友在拜访过后是否会高兴。

输入格式

输入的第一行包含两个整数 N 和 M 。

第二行包含一个长为 N 的字符串。如果第 i 个农场中的奶牛是更赛牛，则字符串中第 i 个字符为 **G**，如果第 i 个农场中的奶牛是荷斯坦牛则为 **H**。

以下 $N - 1$ 行，每行包含两个不同的整数 X 和 Y ($1 \leq X, Y \leq N$)，表示农场 X 与 Y 之间有一条道路。

以下 M 行，每行包含整数 A_i , B_i ，以及一个字符 C_i 。 A_i 和 B_i 表示朋友 i 拜访时行走的路径的端点， C_i 是 **G** 或 **H** 之一，表示第 i 个朋友喜欢更赛牛的牛奶或是荷斯坦牛的牛奶。

输出格式

输出一个长为 M 的二进制字符串。如果第 i 个朋友会感到高兴，则字符串的第 i 个字符为 **1**，否则为 **0**。

样例 #1

样例输入 #1

```
5 5
HHGHG
1 2
2 3
2 4
1 5
1 4 H
1 4 G
1 3 G
1 3 H
5 5 H
```

样例输出 #1

```
10110
```

提示

在这里，从农场 1 到农场 4 的路径包括农场 1、2 和 4。所有这些农场里都是荷斯坦牛，所以第一个朋友会感到满意，而第二个朋友不会。

关于部分分：

测试点 1 样例。

测试点 2 ~ 5 满足 $N \leq 10^3$, $M \leq 2 \cdot 10^3$ 。

对于 100% 的数据, $1 \leq N \leq 10^5$, $1 \leq M \leq 10^5$ 。

B 道

时空限制：1s 256MB

文件名

pump.in/pump.out/pump.cpp

题目描述

Farmer John 最近为了扩张他的牛奶产业帝国而收购了一个新的农场。这一新的农场通过一个管道网络与附近的小镇相连，FJ 想要找出其中最合适的一组管道，将其购买并用来将牛奶从农场输送到小镇。

这个管道网络可以用 N 个接合点（管道的端点）来描述，将其编号为 $1 \dots N$ 。接合点 1 表示 FJ 的农场，接合点 N 表示小镇。有 M 条双向的管道，每条连接了两个接合点。使用第 i 条管道需要 FJ 花费 c_i 美元购入，可以支持每秒 f_i 升牛奶的流量。

FJ 想要购买一条管道组成一条单一路径，路径的两端点分别为接合点 1 和 N 。这条路径的花费等于路径上所有管道的费用之和。路径上的流量等于路径上所有管道的最小流量（因为这是沿这条路径输送牛奶的瓶颈）。FJ 想要最大化路径流量与路径花费之比。保证存在从 1 到 N 之间的路径。

输入格式

输入的第一行包含 N 和 M 。以下 M 行每行以四个整数描述一条管道： a 和 b （管道连接的两个不同的接合点）， c （管道的花费），以及 f （管道的流量）。花费和流量均为范围 $1 \dots 1000$ 之内的正整数。

输出格式

输出 10^6 乘以最优解的值，并向下取整（也就是说，如果这个数本身不是整数，输出小于它的最接近它的整数）。

样例 #1

样例输入 #1

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

样例输出 #1

```
428571
```

提示

在这个例子中，仅由一条路径从 1 到 N 。它的流量为 $\min(3, 4) = 3$ ，花费为 $2 + 5 = 7$ 。

数据范围

测试点 2 ~ 5 满足 $N, M \leq 100$ 。
对于 100% 的数据， $2 \leq N \leq 1000, 1 \leq M \leq 1000$ 。

C 派

时空限制：1s 256MB

文件名

pie.in/pie.out/pie.cpp

题目描述

Farmer John 有 M 头奶牛，为了方便，编号为 $1, \dots, M$ 。这些奶牛平时都吃青草，但是喜欢偶尔换换口味。Farmer John 一天烤了 N 个派请奶牛吃，这 N 个派编号为 $1, \dots, N$ 。第 i 头奶牛喜欢吃编号在 $[l_i, r_i]$ 中的派（包括两端），并且没有两头奶牛喜欢吃相同范围的派。第 i 头奶牛有一个体重 w_i ，这是一个在 $[1, 10^6]$ 中的正整数。

Farmer John 可以选择一个奶牛序列 c_1, c_2, \dots, c_K ，并让这些奶牛按这个顺序轮流吃派。不幸的是，这些奶牛不知道分享！当奶牛吃派时，她会把她喜欢吃的派都吃掉——也就是说，她会吃掉编号在 $[l_{c_i}, r_{c_i}]$ 中所有剩余的派。Farmer John 想要避免当轮到一头奶牛吃派时，她所有喜欢的派在之前都被吃掉了这样尴尬的情况。因此，他想让你计算，要使奶牛按 c_1, c_2, \dots, c_K 的顺序吃派，轮到这头奶牛时她喜欢的派至少剩余一个的情况下，这些奶牛的最大可能体重 $(w_{c_1} + w_{c_2} + \dots + w_{c_K})$ 是多少。

输入格式

第一行包含两个正整数 N, M ；
接下来 M 行，每行三个正整数 w_i, l_i, r_i 。

输出格式

输出对于一个合法的序列，最大可能的体重值。

样例 #1

样例输入 #1

```
2 2
100 1 2
100 1 1
```

样例输出 #1

200

提示

样例解释

在这个样例中，如果奶牛 1 先吃，那么奶牛 2 就吃不到派了。然而，先让奶牛 2 吃，然后奶牛 1 只吃编号为 2 的派，仍可以满足条件。

对于全部数据， $1 \leq N \leq 300, 1 \leq M \leq \frac{N(N-1)}{2}, 1 \leq l_i, r_i \leq N, 1 \leq w_i \leq 10^6$ 。

数据范围

对于测试点 2 – 5，满足 $N \leq 50, M \leq 20$ ；

对于测试点 6 – 9，满足 $N \leq 50$ 。