

实外 CCF CSP2022-S 国庆模拟赛

(提高组：第2场)

时间：2022 年 10 月 5 日 8:00 ~ 12:00

题目名称	改错	网络修复	王位继承	反射镜
题目类型	传统型	传统型	传统型	传统型
目录	correct	network	throne	mirror
可执行文件名	correct	network	throne	mirror
输入文件名	correct.in	network.in	throne.in	mirror.in
输出文件名	correct.out	network.out	throne.out	mirror.out
每个测试点时限	1.0 秒	1.0 秒	2.0 秒	1.0 秒
内存限制	256 MB	256 MB	256 MB	256 MB
子任务数目	20	20	10	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	correct.cpp	network.cpp	throne.cpp	mirror.cpp
对于 C 语言	correct.c	network.c	throne.c	mirror.c
对于 Pascal 语言	correct.pas	network.pas	throne.pas	mirror.pas

编译选项

对于 C++ 语言	-O2
对于 C 语言	-O2
对于 Pascal 语言	-O2

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 `0`。
3. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 程序可使用的栈空间内存限制与题目的内存限制一致。
7. 全国统一评测时采用的机器配置为：Inter(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。上述时限以此配置为准。
8. 只提供 Linux 格式附加样例文件。
9. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

改错 (correct)

【题目描述】

一次，我们的 Mosaic 在很长时间的思考后，终于做出了一道题，结果，……，
WWWWTTTTT，……

被枫哥严厉批评后，Mosaic 开始改错。

经过检查，他发现他的程序中有 N 处错误。而这 N 处错误之间存在 M 条奇怪的的联系。
具体来说，第 i 条联系意味着第 A_i 处错误必须先于第 B_i 处错误被改正。

Mosaic 希望你给出一种字典序最小的改错顺序。

【输入格式】

从文件 **correct.in** 中读入数据。

第一行两个整数 N, M 。

以下 M 行，每行两个整数 A_i, B_i 。

【输出格式】

输出到文件 **correct.out** 中。

一行 N 个整数。表示字典序最小的可行的改错顺序。

【样例输入1】

5 0

【样例输出1】

1 2 3 4 5

【样例输入2】

5 3

2 1

3 2

4 5

【样例输出2】

3 2 1 4 5

【数据范围】

$0 \leq n \leq 1000, 0 \leq m \leq 1000$ 。保证至少存在一组解。

网络修护 (network)

【题目描述】

经过同学们的改造，信奥班的网络成为了一棵有 N 个节点的无根树，每一个节点代表一台计算机（编号 $1 \sim N$ ），每一条边代表一条网络线路（编号 $1 \sim (N - 1)$ ）。

有一天，Gemini 偶然发现了 Kroulis 以前写过一个杀毒软件；于是他毫不犹豫地找到了病毒库，运行了里面所有的病毒。（我：什么心态？Gemini：出题需要嘛！我：不对怎么精分现场了？Gemini：怪我咯？！）

于是这一下，每一条网络线路都断开了。在收拾了 Gemini 后，同学们开始抢修网络。抢修过程持续了 T 个时刻，在每一时刻有可能发生下面两件事之一：

1. 某一条网络线路恢复通畅；
2. Gemini 想知道某两台计算机之间是否已经连通，如果连通，从一台计算机到另一台计算机要经过多少条网络线路。

【输入格式】

从文件 **network.in** 中读入数据。

第一行两个整数 N, T 。

以下 $N - 1$ 行每行两个整数 A_i, B_i ，表示第 i 条网络线路直接连接计算机 A_i 和 B_i 。

以下 T 行顺序给出 T 个事件，每行形如 “C e” 或 “Q a b”（其中 e, a, b 是整数）。“C e” 代表这一时刻第 e 条网络线路恢复通畅；“Q a b” 代表查询第 a 台计算机和第 b 台计算机之间是否连通。

【输出格式】

输出到文件 **network.out** 中。

为了避免大量输出，你需要按照如下方式处理你的输出。

声明一个 32 位有符号整型变量 R ，赋值为 0。将所有事件按照发生顺序（即输入文件中给出的顺序）编号为 $1 \sim T$ 。按以下指示遍历所有操作。

1. 如果第 i 个事件为 “C e”，则不对 R 做任何处理。
2. 如果第 i 个事件为 “Q a b”，且 a, b 不连通，则 $R = R \wedge (i + a + b)$ 。
3. 如果第 i 个事件为 “Q a b”，且 a, b 连通，且从计算机 a 到计算机 b 要经过 K 条网络线路，则 $R = (R + (i \wedge K)) \% 998244353$ 。

最终输出 R 的值。（ \wedge 代表二进制下的异或操作， $\%$ 代表取模操作。）

【样例输入】

2 3

1 2

Q 1 2

C 1

Q 2 1

【样例输出】

6

【数据范围】

$0 \leq N, T \leq 100000$ 。

$1 \leq a, b \leq n \leq N, 1 \leq e \leq N-1$ 。

王位继承 (throne)

【题目描述】

Airy 是信奥班的萌神三氧化萌 (MeO3, moe trioxide)。爱学习的 Airy 有一天钻研起了英国的王位继承顺序。(因为 Airy 萌呀 233333333)
假设写成伪代码的话，英国的王位继承顺序是这样的：

```
solve(某王室成员 x) {  
    把 x 加入继承序列  
    按年龄从大到小遍历 x 的所有儿子 i : solve(i)  
    按年龄从大到小遍历 x 的所有女儿 i : solve(i)  
}  
  
main() {  
    solve(国王/女王)  
}
```

注意此处把国王/女王也加入了继承序列，并且排在首位。我们把国王/女王本人定义为第 0 顺位继承人。假设王室成员一共有 N 个人（包括国王/女王），我们把继承序列里其余 $(N - 1)$ 个人按加入继承序列的先后顺序定义为第 1 顺位继承人到第 $(N - 1)$ 顺位继承人。

我们这样定义“后代”这个概念：

1. A 的所有儿女是 A 的后代；
2. 如果 B 是 A 的后代，那么 B 的所有后代都是 A 的后代。

Airy 找到了所有 N 个继承人的信息，包括每个继承人的继承顺序、性别和后代个数。现在她把这些信息全部给你。为什么会给你呢？因为 Airy 有事情要找你帮忙。（我：是不是什么奇怪的事情啊？Airy：= =你说什么只是出题需要啦……）

原来，在某一年英国改变了原有的王位继承顺序：取消了原有的男性优于女性的原则，在排序时不再将儿子、女儿分开，而是一视同仁。因此，假设还是写成伪代码的话，现在的伪代码成了这样：

```

solve(某王室成员 x) {
    把 x 加入继承序列
    按年龄从大到小遍历 x 的所有儿女 i : solve(i)
}

main() {
    solve(国王/女王)
}

```

现在，Airy 想知道，改革后的第 M 顺位继承人在改革前后继承顺序变化了多少（定义上升为正，下降为负）。

【输入格式】

从文件 **throne.in** 中读入数据。

第一行两个整数 N, M。

以下 N 行，每行两个整数 A_i B_i ，为第 i 个继承人的信息。 A_i 为第 i 个继承人的后代个数； B_i 为第 i 个继承人的性别（0：女，1：男）。当然，Airy 给你的数据都是改革后的。

【输出格式】

输出到文件 **throne.out** 中。

一行一个整数，为得到的答案。

【样例输入1】

```

3 1
2 1
0 0
0 1

```

【样例输出1】

```

1

```

【样例输入2】

```

1 0
0 0

```

【样例输出2】

```

0

```

【数据范围】

对于40%的数据， $1 \leq N \leq 10000$ 。

对于100%的数据， $1 \leq N \leq 1000000$ 。

$1 \leq M < N$ ，保证每个王室成员的子女不会超过 10 个人。保证王室不会超过 1,000,000 代人。

反射镜 (mirror)

【题目描述】

从前有一个坐标网格（其中坐标的绝对值不会超过 m ）。从左到右 x 坐标逐渐增加，而从下到上 y 坐标逐渐增加。

在网格中摆放着 n 面镜子，第 i 面镜子的坐标为 (x_i, y_i) 。镜子均与坐标轴成 45° 角。所以共有两种类型的镜子：“\”型和“/”型。特殊地，原点处不会有任何镜子，也不会有某个位置有多面镜子。

镜子的两个面都能够反射光线，而中间不透光，例如，对于一个“/”型镜子，从下面射入的光线会被反射到右方向，而从左面射入的光线会被反射到上方向。

现有一条光线从原点所在格子沿 x 轴正方向射出，求它走过 T 格路程后所在的位置。

【输入格式】

从文件 **mirror.in** 读入数据。

第一行为三个正整数 n, m, T ，分别代表镜子的个数与坐标的范围。

接下来 n 行，每行两个整数和一个字符 x_i, y_i, t_i ，代表某个镜子的位置和类型。

【输出格式】

输出到文件 **mirror.out** 中。

输出两个整数，分别代表走过 T 格路程后的 x 坐标和 y 坐标。

【样例输入】

```
5 2 8
0 1 \
0 2 /
1 0 /
1 1 \
1 2 \
```

【样例输出】

```
3 1
```

【样例解释】

在8个单位时间的路线是：右上左上右下右右。

【数据规模与约定】

存在10%的数据， $n = 1$ 。
存在40%的数据， $n \leq 1,000$ 。
存在40%的数据， $m \leq 1,000$ 。
存在40%的数据， $T \leq 1,000,000$ 。
对于100%的数据， $n \leq 100,000, m \leq 1,000,000,000, T \leq 10^{18}$ 。
具体地，数据范围如下表所示：

测试点编号	n	m	T
1	$= 1$	$\leq 1,000$	$\leq 100,000$
2,3		$\leq 1,000$	$\leq 100,000$
4,5	$\leq 1,000$		$\leq 100,000$
6,7	$\leq 1,000$	$\leq 1,000$	
8	$= 1$		
9,10	$\leq 1,000$		
11,12,13		$\leq 1,000$	
14,15,16			$\leq 100,000$
17,18,19,20			