

全国青少年信息学奥林匹克竞赛

NOI2023模拟

时间：7:30-12:20

题目名称	礼物	镜子游戏	序列
题目类型	传统型	传统型	传统型
目录	gift	mirror	sequence
可执行文件名	gift	mirror	sequence
输入文件名	gift.in	mirror.in	sequence.in
输出文件名	gift.out	mirror.out	sequence.out
每个测试点时限	1.0秒	2.0秒	1.0秒
内存限制	512 MB	1024 MB	512MB
子任务数目	25	20	25
测试点是否等分	是	是	是

提交源程序文件名

对于C++语言	gift.cpp	mirror.cpp	sequence.cpp
---------	----------	------------	--------------

编译选项

对于C++语言	-lm -std=c++14 -O2
---------	--------------------

注意事项与提醒（请选手务必仔细阅读）

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++ 中主函数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照各省的具体要求。
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. 程序可使用的栈内存空间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：Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz，内存 32GB。上述时限以此配置为准。
8. 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其为准。
9. 终评测时所用的编译命令中不含编译选项之外的任何优化开关。

礼物 (gift)

【问题描述】

OI 国有 n 座城市，其中 1 号城市是首都。这些城市被一些 **单向** 公路连接，对于城市 $i \neq 1$ ，有一条线路从 i 到 p_i ($p_i < i$)。每一条线路都是一样长的，通行花费的时间也是一样的。

五湖四海的 OIer (总共 c 人, $2 \leq c \leq 5$) 即将在某个城市进行 NOI 前的最后集训，为了节省赶路的时间，他们希望集训的城市满足每个人从各自城市同时出发能 **尽快到齐**。（注意，**可能有多个人在同一个城市**）

许多 OIer 是初次见面，为了加深各自的友谊，他们决定沿途购买礼物用以互相赠送。OI 国家的每一个城市都有一种特产，整个国家共有 m 种特产，其中城市 i 的特产用 a_i 表示（不同城市可能有相同的特产）。

购买礼物时，必须满足以下条件：

1. 每个人带的礼物数量必须相同
2. 到达集训城市时，所有礼物中不能够有任何两个类别相同。
3. 每个人只能带他所经过的城市的特产作为礼物。

现在有多个询问，每次给出 OIer 的不同位置分布，要求对于每个询问，计算出集训时最多有多少种礼物。

【输入格式】

从输入文件 gift.in 中读入数据；

第一行有三个整数 n, m, q ，分别表示城市个数，特产种数，询问个数；

第二行有 $n - 1$ 个整数，表示 p_2, p_3, \dots, p_n 。

第三行有 n 个整数，表示 a_1, a_2, \dots, a_n 。

接下来 q 行，每行表示一个询问。每个询问第一个整数 c 表示人数，接下来有 c 个整数表示每一个人所在城市编号。

【输出格式】

输出到文件 party.out 中。

对于每个询问输出一行一个整数，表示答案。

【样例输入1】

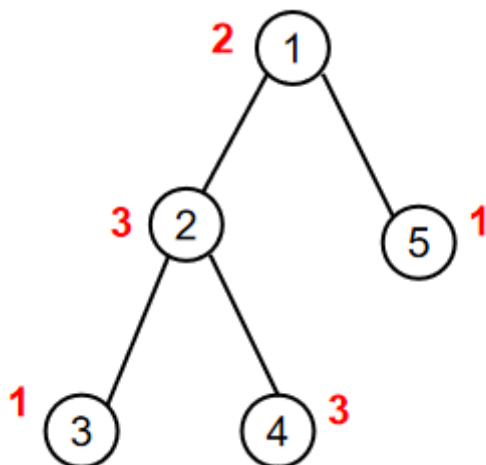
```
5 3 4
1 2 2 1
2 3 1 3 1
2 3 4
3 5 2 2
4 3 4 2 5
2 2 2
```

【样例输出1】

```
2
3
0
0
```

【样例1解释】

城市如下图所示：



对于询问一：一种最优的方案是选择 2 号城市作为集训地点，且 1 号同学购买 1 号特产，2 号同学购买 3 号特产。

对于询问二：一种最优的方案是选择 1 号城市作为集训地点，且 1 号同学购买 1 号特产，2 号同学购买 2 号特产，3 号同学购买 3 号特产。

对于询问三：

如果选择 2 号城市作为集训地点：由于 2 号同学只能购买 3 号特产，那么 4 号同学无法购买特产，因此没有可行方案。

如果选择 1 号城市作为集训地点：由于 2, 3, 4 号同学只能购买 1, 2, 3 号特产，4 号同学只能购买 1, 2 号特产，因此没有可行方案。

对于询问四：只能选择 2 号城市作为集训地点，此时两个同学没有可行方案。

【样例2】

见选手目录下的 *ex_gift2.in* 与 *ex_gift2.ans*。

【数据范围及约定】

对于 100% 的数据，满足 $2 \leq n \leq 300000$, $1 \leq m \leq 1000$, $0 \leq q \leq 50000$, $1 \leq p_i < i$, $1 \leq a_i \leq m$, $2 \leq c \leq 5$ 。

测试点编号	n	q	其他约定
1, 2	300000	0	无
3, 4, 5	10	10	无
6, 7, 8	3000	3000	$c = 2$
9, 10, 11	3000	3000	$c \leq 3$
12, 13, 14, 15, 16	300000	3000	无
17, 18, 19	300000	50000	$p_i = i - 1$
20, 21, 22, 23, 24, 25	300000	50000	无

镜子游戏 (mirror)

【问题描述】

约翰·李最近沉迷于益智类游戏，以期望在NOI赛场上能够提升智力，突破构造题目。

最近他发现了一个简化版的镜子游戏，游戏的场景是一个 $n \times n$ 的网格图，其中第 i 行第 j 列的格子记作 (i, j) 。每个格子可能是空的或者障碍，恰好有 m 个障碍。

现在有一束光线从 $(1, 1)$ 沿着 $(2, 1), (3, 1), \dots$ 方向纵向射入。光线会沿着空格子射过去，但当遇到障碍或者边界就会停下来。

现在，约翰可以在某些格子上放置镜子，使得光线到达镜子处会转向 90 度反射出去（每个格子至多放置一个镜子）。

游戏目标是让光线通过若干次反射能够到达某个位置的出口，为了能够快速通关游戏，约翰希望能够提前知晓每个位置作为出口至少需要放置几个镜子。

即，记 $f(x, y)$ 表示光线从 $(1, 1)$ 到 (x, y) 至少要放几个镜子。如果这个格子是障碍或者不可达，那么就记 $f(x, y) = 0$ ，问所有格子的 f 值之和。

【输入格式】

第一行两个整数 n, m 。

接下来 m 行，每行两个整数 $x, y (1 \leq x, y \leq n)$ 障碍的位置，保证障碍的位置两两不同，并且 $(1, 1)$ 格子不为障碍。

【输出格式】

输出一行一个整数表示答案。

【样例输入1】

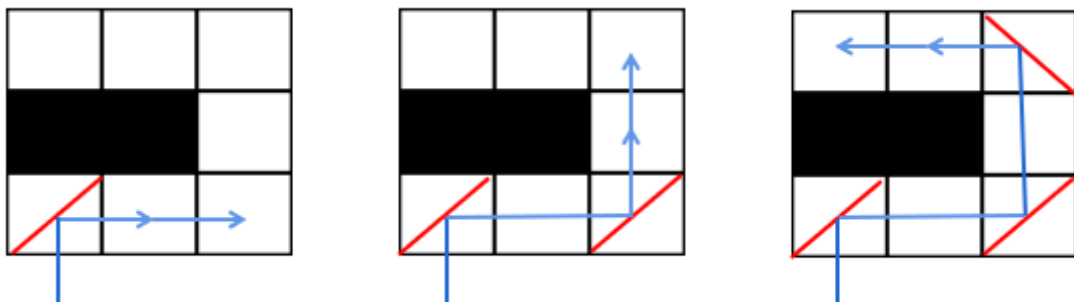
```
3 2
2 2
2 1
```

【样例输出1】

```
12
```

【样例1解释】

如下图所示， $f(1,1)=f(2,1)=f(2,2)=0$ ， $f(1,2)=f(1,3)=1$ ， $f(2,3)=f(3,3)=2$ ， $f(3,2)=f(3,1)=3$



【样例2】

见选手目录下的 *ex_mirror2.in* 与 *ex_mirror2.ans*。

【数据范围及约定】

测试点	$n \leq$	$m \leq$
1, 2, 3, 4	100	100
5, 6, 7, 8	1000	1000
9, 10, 11, 12	10^5	1000
13, 14, 15, 16, 17, 18, 19, 20	10^5	10^5

序列 (sequence)

【问题描述】

约翰·李和一念之间最近学习了一个经典的NIM问题：一开始有 n 堆石头，第 i 堆石头有 a_i 个，两个玩家轮流操作，每次选最多两堆石头，拿走里面任意颗石头（加起来至少一颗），不能操作的玩家输。

但是一念之间对于题目的思维难度并不满意，决定出一道题目的强化版本：现在给定 l_1, \dots, l_n 和 r_1, \dots, r_n ，你要求有几个 a_1, \dots, a_n 序列满足：

1. 对于所有 i 有 $l_i \leq a_i \leq r_i$
2. a_1, \dots, a_n 满足后手必胜

由于答案可能很大，你只需要输出答案对 998244353 取模后的值。

【输入格式】

第一行一个正整数 n

接下来 n 行，每行两个整数 l_i, r_i 。

【输出格式】

输出一行仅一个整数，表示答案。

【样例输入1】

```
4
0 2
0 2
0 2
0 3
```

【样例输出1】

```
9
```

【样例1解释】

满足条件的 a 序列有如下 9 种：(0, 0, 0, 0), (1, 1, 1, 0), (1, 1, 0, 1), (1, 0, 1, 1), (0, 1, 1, 1), (2, 2, 2, 0), (2, 2, 0, 2), (2, 0, 2, 2), (0, 2, 2, 2)。

【样例2】

见选手目录下的 `ex_sequence2.in` 与 `ex_sequence2.ans`。

【数据范围及约定】

对于所有数据，有 $2 \leq n \leq 10$, $0 \leq l_i \leq r_i < 2^{30}$ 。

测试点	n	r_i
1, 2, 3	≤ 5	≤ 5
4, 5, 6, 7, 8, 9, 10	≤ 10	< 128
11, 12, 13, 14, 15, 16	≤ 10	$< 2^{11}$
17, 18, 19, 20	≤ 6	$< 2^{30}$
21, 22, 23, 24, 25	≤ 10	$< 2^{30}$