全国青少年信息学奥林匹克联赛模拟赛

by GDSY

模拟赛

题目名称	Soso 的期望并查集	Soso 的模法矩阵	Soso 的排列	走路
题目类型	传统型	传统型	传统型	传统型
目录	`exsodsu`	`modmat`	`soperme`	`walk`
可执行文件名	`exsodsu`	`modmat`	`soperme`	`walk`
输入文件名	`exsodsu.in`	`modmat.in`	`soperme.in`	`walk.in`
输出文件名	`exsodsu.out`	`modmat.out`	`soperme.out`	`walk.out`
每个测试点时限	2.0 秒	2.0 秒	2.0 秒	3.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	8	10	8	8
测试点是否等分	否	否	否	否

提交源程序文件名

对于 C++ 语言	`exsodsu.cpp`	`modmat.cpp`	`soperme.cpp`	`walk.cpp`

编译选项

对于 C++ 语言	`-O2 -std=c++14 -static`

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. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 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。
9. 只提供 Linux 格式附加样例文件。
10. 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。

Soso 的期望并查集（exsodsu）

【题目描述】

Soso 欲将实现一个并查集，但是他写挂了。具体来说，他有 n 个点，第 i 个点的权值为 a_i，初始时每个点都是一棵有根树。

定义操作 $\text{find}(x)$，是找到 x 所在有根树的根。这次操作的代价是 x 到根的路径上所有点的点权和。

定义操作 $\text{merge}(x, y)$：

- 首先找到 x, y 所在有根树的根，记 $x' = \text{find}(x), y' = \text{find}(y)$。
- 若 $x' \neq y'$，将 y' 的父亲设为 x'。这意味着 x', y' 这两棵有根树合并到一起，以后 $\text{find}(y)$ 和 $\text{find}(x)$ 应该相等。

明显，这个并查集的实现过程过于暴力，Soso 想计算一下它到底有多么暴力。给出 m 次 $\text{merge}(x, y)$ 的操作，请求出所有操作中 find 产生的代价总和，记为 ans，对 998244353 取模。

但是 Soso 要加强这个题。

每一次 merge 操作时，Soso 会额外给一个概率 p，表示有 p 的概率，宇宙射线会影响这个程序，使得 x, y 的值交换；有 $1 - p$ 的概率，x, y 不交换。

现在 Soso 要你求出宇宙射线干扰之后 ans 的期望值。

【输入格式】

从文件 ***exsodsu.in*** 中读入数据。

第一行两个正整数 n, m（$n, m \leq 5 \times 10^5$），表示有 n 个点 m 次操作。

第二行 n 个非负整数 a_i 表示点权（$0 \leq a_i < 998244353$）。

接下来 m 行，每行两个正整数 x, y（$1 \leq x, y \leq n$）和一个非负整数 p（$0 \leq p < 998244353$），表示执行操作 $\text{merge}(x, y)$，但是有 p 的概率交换 x, y。p 应该是一个区间 $[0, 1]$ 内的有理数，为了方便，输入了 $p \bmod 998244353$ 的结果。

【输出格式】

输出到文件 ***exsodsu.out*** 中。

一行一个整数表示所有操作中 $\text{find}(x)$ 产生的代价总和 ans 的期望，对 998244353 取模。

【样例 1 输入】

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

```
2 5 499122177
4 3 0
5 1 0
2 2 0
3 5 0
```

【样例 1 输出】

```
38
```

【样例 1 解释】

样例的 499122177 取模前是 $\frac{1}{2}$。若第一次操作没有发生交换，则 ans 计算如下：

- $\text{merge}(2,5)$
  - $\text{find}(2)=2$，代价为 5。
  - $\text{find}(5)=5$，代价为 3。
  - 将 5 的父亲设为 2。
- $\text{merge}(4,3)$
  - $\text{find}(4)=4$，代价为 1。
  - $\text{find}(3)=3$，代价为 1。
  - 将 3 的父亲设为 4。
- $\text{merge}(5,1)$
  - $\text{find}(5)=2$，代价为 $3+5=8$。
  - $\text{find}(1)=1$，代价为 2。
  - 将 1 的父亲设为 2。
- $\text{merge}(2,2)$
  - $\text{find}(2)=2$，代价为 5。
  - $\text{find}(2)=2$，代价为 5。
  - 无操作。
- $\text{merge}(3,5)$
  - $\text{find}(3)=4$，代价为 $1+1=2$。
  - $\text{find}(5)=2$，代价为 $3+5=8$。
  - 将 2 的父亲设为 4。
- 算法结束时，记 fa_i 为 i 的父亲，则 $fa=\{2,4,4,\varnothing,2\}$，四号点是树根。

- 将上面的代价全部相加得到答案是 40。

若第一次操作发生了交换则 ans 计算如下：

- $\text{merge}(5,2)$
  - $\text{find}(5)=5$，代价为 3。
  - $\text{find}(2)=2$，代价为 5。
  - 将 2 的父亲设为 5。
- $\text{merge}(4,3)$
  - $\text{find}(4)=4$，代价为 1。
  - $\text{find}(3)=3$，代价为 1。
  - 将 3 的父亲设为 4。
- $\text{merge}(5,1)$
  - $\text{find}(5)=5$，代价为 3。
  - $\text{find}(1)=1$，代价为 2。
  - 将 1 的父亲设为 5。
- $\text{merge}(2,2)$
  - $\text{find}(2)=5$，代价为 $5+3=8$。
  - $\text{find}(2)=5$，代价为 $5+3=8$。
  - 无操作。
- $\text{merge}(3,5)$
  - $\text{find}(3)=4$，代价为 $1+1=2$。
  - $\text{find}(5)=5$，代价为 3。
  - 将 5 的父亲设为 4。
- 算法结束时，记 fa_i 为 i 的父亲，则 $fa=\{5,5,4,\varnothing,4\}$，四号点是树根。
- 将上面的代价全部相加得到答案是 36。

根据期望的定义，得到 ans 的期望是 $40\times\frac{1}{2}+36\times\frac{1}{2}=38$。

【样例 2】

见选手目录下的 ***exsodsu/exsodsu2.in*** 与 ***exsodsu/exsodsu2.ans***。

【样例 3】

见选手目录下的 ***exsodsu/exsodsu3.in*** 与 ***exsodsu/exsodsu3.ans***。

【样例 4】

见选手目录下的 ***exsodsu/exsodsu4.in*** 与 ***exsodsu/exsodsu4.ans***。

【样例 5】

见选手目录下的 ***exsodsu/exsodsu5.in*** 与 ***exsodsu/exsodsu5.ans***。

【子任务】

对于所有数据，$n,m \le 5\times 10^5$，$0 \le a_i,p < 998244353$，$1 \le x,y \le n$。

子任务	特殊性质	分数	依赖于
1	$n,m \le 10$	10	
2	$n,m \le 100$	10	1
3	$n,m \le 1000$	10	2
4	$n,m \le 10^5$	20	3
5	$p=0$	10	
6	$\sum_{i=1}^{n} a_i = 1$	10	
7	最多只有 5 个 p 非零	10	5
8	无特殊限制	20	$4,6,7$

Soso 的模法矩阵（modmat）

【题目描述】

Soso 是你的数学老师。今天 Soso 想到了一个题目：给定长为 n 的正整数数组 $\{a_i\}$ 与长为 m 的正整数数组 $\{b_i\}$。对每对 $i_0 \in [1,n], j_0 \in [1,m]$ 求出

$$\left(\prod_{i=1}^{i_0} a_i\right) \bmod \left(\prod_{j=1}^{j_0} b_j\right)$$

对 $p = 998244353$ 取模的结果，记为 $f(i_0, j_0)$。

这明显是一道 OI 题，不是 Soso 所擅长的领域，所以就只能你来做了。

【输入格式】

从文件 ***modmat.in*** 中读入数据。

第一行两个正整数 n, m（$n, m \le 5000$）。

第二行 n 个正整数，第 i 个是 a_i（$1 \le a_i \le 10^9$）。

第三行 m 个正整数，第 i 个是 b_i（$1 \le b_i \le 10^9$）。

【输出格式】

输出到文件 ***modmat.out*** 中。

由于输出量过大，你需要对答案做一些处理。

输出 n 行，第 i 行输出一个非负整数为下式：

$$\sum_{j=1}^{m} f(i,j) \times p^{m-j} \pmod{10^9+7}$$

【样例 1 输入】

```
4 5
11 7 27 6
2 3 3 4 5
```

【样例 1 输出】

```
984521671
69378816
999420803
968398469
```

【样例 1 解释】

以下第 i 行第 j 个数为 $f(i,j)$。

```
1 5 11 11 11
1 5 5 5 77
1 3 9 63 279
0 0 0 18 234
```

【样例 2 输入】

```
10 10
2 19 23 2 5 19 16 4 2 5
11 17 4 13 30 15 29 4 6 30
```

【样例 2 输出】

```
796095607
398854465
610137974
297291944
145820972
279127363
850690159
47413999
782025003
979354020
```

【样例 2 解释】

以下第 i 行第 j 个数为 $f(i,j)$。

```
2 2 2 2 2 2 2 2 2 2
5 38 38 38 38 38 38 38 38 38
5 126 126 874 874 874 874 874 874 874
10 65 252 1748 1748 1748 1748 1748 1748 1748
6 138 512 8740 8740 8740 8740 8740 8740 8740
4 4 4 752 166060 166060 166060 166060 166060 166060
9 64 64 2308 31480 2656960 2656960 2656960 2656960 2656960
```

```
3 69 256 9232 125920 1876240 10627840 10627840 10627840 10627840
6 138 512 8740 251840 3752480 21255680 21255680 21255680 21255680
8 129 316 4804 92320 1259200 106278400 106278400 106278400
  106278400
```

【样例 3】

见选手目录下的 ***modmat/modmat3.in*** 与 ***modmat/modmat3.ans***。
该样例满足子任务 5 的性质且额外满足 $n, m \le 1000$。

【样例 4】

见选手目录下的 ***modmat/modmat4.in*** 与 ***modmat/modmat4.ans***。
该样例满足子任务 7 的性质。

【样例 5】

见选手目录下的 ***modmat/modmat5.in*** 与 ***modmat/modmat5.ans***。
该样例满足子任务 8 的性质。

【样例 6】

见选手目录下的 ***modmat/modmat6.in*** 与 ***modmat/modmat6.ans***。
该样例满足子任务 9 的性质。

【样例 7】

见选手目录下的 ***modmat/modmat7.in*** 与 ***modmat/modmat7.ans***。
该样例满足子任务 10 的性质。

【样例 8】

见选手目录下的 ***modmat/modmat8.in*** 与 ***modmat/modmat8.ans***。
该样例满足子任务 10 的性质。

【子任务】

对于所有数据：$1 \le n, m \le 5000$，$1 \le a_i, b_j \le 10^9$。下面表格中正整数 i, j 的值域分别为 $[1, n]$、$[1, m]$。

<table>
<tr><th>子任务</th><th>$n, m \le$</th><th>a_i</th><th>b_j</th><th>分数</th><th>依赖于</th></tr>
<tr><td>1</td><td>$2,500$</td><td>$< p$</td><td>$= a_i$</td><td>5</td><td></td></tr>
<tr><td>2</td><td>15</td><td>≤ 15</td><td>≤ 15</td><td>5</td><td></td></tr>
<tr><td>3</td><td>70</td><td rowspan="7">$< p$</td><td rowspan="3">$= 10$</td><td>10</td><td></td></tr>
<tr><td>4</td><td>400</td><td>15</td><td>3</td></tr>
<tr><td>5</td><td>$2,500$</td><td>15</td><td>4</td></tr>
<tr><td>6</td><td>70</td><td rowspan="4">$< p$</td><td>5</td><td>$2,3$</td></tr>
<tr><td>7</td><td>400</td><td>5</td><td>$4,6$</td></tr>
<tr><td>8</td><td>$2,500$</td><td>20</td><td>$1,5,7$</td></tr>
<tr><td>9</td><td rowspan="2">$5,000$</td><td>15</td><td>8</td></tr>
<tr><td>10</td><td>$\le 10^9$</td><td>$\le 10^9$</td><td>5</td><td>9</td></tr>
</table>

注意：子任务 1 满足 $n = m$ 的限制。

Soso 的排列（soperme）

【题目背景】

n 阶排列是一个包含 n 个正整数的数组 $p = \{p_1, p_2, \cdots, p_n\}$，满足 $1 \le p_i \le n$，同时不存在 $1 \le i < j \le n$ 使得 $p_i = p_j$。

n 阶排列 p 的字典序比 n 阶排列 q 的字典序小，当且仅当存在一个正整数 $1 \le pos \le n$ 使得对于所有 $1 \le i < pos$ 都有 $p_i = q_i$ 且 $p_{pos} < q_{pos}$。

n 阶排列 p 的字典序比 n 阶排列 q 的字典序大，当且仅当 q 的字典序比 p 的字典序小。

【题目描述】

Soso 有一个 n 阶排列 p。Soso 将对这个排列做 k 次 `std::next_permutation`，每做一次 `std::next_permutation` 就将当前排列累加到另外一个数组 s 中。

具体来说，初始时 s 为 n 个 0 组成的数组，每次将 p 修改成比 p 字典序大的 n 阶排列中字典序最小的那一个（如果不存在这样的排列，将所有 p_i 分别修改为 i），然后对所有 $1 \le i \le n$，将 s_i 修改为 $s_i + p_i$。

做完 k 次操作后，Soso 想知道 s 数组的具体数值，可是他的暴力跑不出来了，于是求助于你。

【输入格式】

从文件 ***soperme.in*** 中读入数据。

第一行三个正整数 id, n, k（$1 \le id \le 8$，$1 \le n \le 2 \times 10^5, 1 \le k \le 10^{12}$）。其中 id 是子任务编号，具体见下。

第二行 n 个正整数，第 i 个数表示 p_i。保证 p 是一个 n 阶排列。

【输出格式】

输出到文件 ***soperme.out*** 中。

输出 n 行，每行一个整数表示 s_i。

【样例 1 输入】

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

【样例 1 输出】

```
5
10
21
20
19
```

【样例 1 解释】

Soso 将这些排列累加得到了答案：

- $(1,2,3,5,4)$
- $(1,2,4,3,5)$
- $(1,2,4,5,3)$
- $(1,2,5,3,4)$
- $(1,2,5,4,3)$

【样例 2 输入】

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

【样例 2 输出】

```
2
4
6
8
10
12
14
16
19
19
```

【样例 2 解释】

Soso 将这些排列累加得到了答案：

- $(1,2,3,4,5,6,7,8,9,10)$
- $(1,2,3,4,5,6,7,8,10,9)$

注意到 `std::next_permutation` 如果找不到下一个排列，会将 p 修改成字典序最小的排列。

【样例 3 输入】

```
1 5 10
2 1 3 5 4
```

【样例 3 输出】

```
20
22
38
35
35
```

【样例 3 解释】

Soso 将这些排列累加得到了答案：

- $(2,1,4,3,5)$
- $(2,1,4,5,3)$
- $(2,1,5,3,4)$
- $(2,1,5,4,3)$
- $(2,3,1,4,5)$
- $(2,3,1,5,4)$
- $(2,3,4,1,5)$
- $(2,3,4,5,1)$
- $(2,3,5,1,4)$
- $(2,3,5,4,1)$

【样例 4】

见选手目录下的 *soperme/soperme4.in* 与 *soperme/soperme4.ans*。

该样例满足 $id=3$。

【样例 5】

见选手目录下的 ***soperme/soperme5.in*** 与 ***soperme/soperme5.ans***。
该样例满足 $id = 4$。

【样例 6】

见选手目录下的 ***soperme/soperme6.in*** 与 ***soperme/soperme6.ans***。
该样例满足 $id = 5$。

【样例 7】

见选手目录下的 ***soperme/soperme7.in*** 与 ***soperme/soperme7.ans***。
该样例满足 $id = 6$。

【样例 8】

见选手目录下的 ***soperme/soperme8.in*** 与 ***soperme/soperme8.ans***。
该样例满足 $id = 7$。

【样例 9】

见选手目录下的 ***soperme/soperme9.in*** 与 ***soperme/soperme9.ans***。
该样例满足 $id = 7$。

【样例 10】

见选手目录下的 ***soperme/soperme10.in*** 与 ***soperme/soperme10.ans***。
该样例满足 $id = 8$。

【子任务】

对于所有数据，保证 $1 \le id \le 8, 1 \le n \le 2 \times 10^5, 1 \le k \le 10^{12}$，$p$ 是 n 阶排列。

子任务	$n \le$	$k \le$	特殊性质	分数	依赖于
1	2×10^5	$10^6/n$	无	10	
2		10^{12}	ABD	5	
3	200	5×10^8	BC	25	
4	2×10^5	10^{12}	B	10	$2, 3$
5	200	5×10^8	AD	20	
6	2×10^5	10^{12}	A	5	$2, 5$
7	500	10^{10}	无	15	$3, 5$
8	2×10^5	10^{12}		10	$1, 4, 6, 7$

特殊性质 A：保证 Soso **初始**拿到的排列 p 满足 $p_i = n - i + 1$。

特殊性质 B：保证 Soso **最终**获得的排列 p 满足 $p_i = n - i + 1$。

特殊性质 C：保证调用 `std::next_permutation` 后返回值**都**为真，即 k 次操作中总是存在一个 n 阶排列使得它的字典序比 p 大。

特殊性质 D：保证**有且仅有一次**调用 `std::next_permutation` 返回值**不**为真，即 k 次操作中有且仅有一次操作使得不存在一个 n 阶排列使得它的字典序比 p 大。

走路（walk）

【题目描述】

Z 市有 n 个路口，$n-1$ 条道路连接这些路口。第 i 个路口有 c_i 条道路以其为端点，按逆时针方向分别直接到达路口 $a_{i,0}, a_{i,1}, \ldots a_{i,c_i-1}$，每个路口可以通过道路直接或间接到达所有路口。每个路口有个可以转的路标，初始指向去往 a_{i,t_i} 的道路。

Alice 计划从路口 1 出发，走过若干条道路。由于她没有 Z 市的地图，她会使用右手定则行动。假设她位于路口 x，她会先让这里的路标转至其逆时针方向的下一条道路，即 $t'_x = (t_x + 1) \bmod c_x$，然后沿着**转向前**路标指向的道路前进，到达 a_{x,t_x}。当她再次回到 x 时，就会去往 a_{x,t'_x}，并令 $t''_x = (t'_x + 1) \bmod c_x$。巧合的是，她的朋友 Bob 有地图，如果你能回答如下问题，他就会把地图交给 Alice。

q 次操作，每次操作形如：

`1 x y`，表示把 t_x 设为 y。

`2 k`，假设 Alice 从 1 出发，求她连续走过 k 条道路后到达的点。操作 2 互相独立，询问后会将所有路标复原。

【输入格式】

从文件 ***walk.in*** 中读入数据。

第一行一个正整数 n，表示路口个数。

接下来 n 行，每行前两个整数 c_i, t_i，接下来 c_i 个整数 $a_{i,0}, a_{i,1}, \ldots a_{i,c_i-1}$。

接下来一行一个正整数 q，表示操作次数。

接下来 q 行，每行表示一个操作，格式见题目描述。

【输出格式】

输出到文件 ***walk.out*** 中。

对于每个操作 2，输出一行一个整数，表示答案。

【样例 1 输入】

```
5
2 0 2 3
3 2 1 4 5
1 0 1
1 0 2
1 0 2
5
```

```
2 5
1 2 0
2 6
1 1 1
2 7
```

【样例 1 输出】

```
3
4
2
```

【样例 1 解释】

三条路径分别为：

- $1 \to 2 \to 5 \to 2 \to 1 \to 3$；
- $1 \to 2 \to 1 \to 3 \to 1 \to 2 \to 4$；
- $1 \to 3 \to 1 \to 2 \to 1 \to 3 \to 1 \to 2$。

【样例 2】

见选手目录下的 ***walk/walk2.in*** 与 ***walk/walk2.ans***。
此样例满足子任务 2 的性质。

【样例 3】

见选手目录下的 ***walk/walk3.in*** 与 ***walk/walk3.ans***。
此样例满足子任务 6 的性质。

【子任务】

对于所有数据，$1 \le n, q \le 10^5, 1 \le c_i, a_{i,j} \le n, 0 < t_i < c_i, 1 \le x \le n, 0 < y < c_x, 1 \le k \le 10^{18}$。

子任务	$n,q\le$	特殊性质	分数	依赖于
1	10	无	8	
2	400		12	1
3	7000		12	2
4	10^5	AB	8	
5		A	8	4
6		B	12	4
7	7×10^4	无	20	3
8	10^5		20	$5,6,7$

特殊性质 A：$c_1=1,c_i\le 2$。

特殊性质 B：$k>10^{12}$。