PION 2024 模拟赛

HEZ

时间：2024 年 11 月 14 日 08:30 ~ 13:00

题目名称	城市	烟火	道路	合唱
题目类型	传统型	传统型	传统型	传统型
目录	`city`	`show`	`road`	`sing`
可执行文件名	`city`	`show`	`road`	`sing`
输入文件名	`city.in`	`show.in`	`road.in`	`sing.in`
输出文件名	`city.out`	`show.out`	`road.out`	`sing.out`
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
子任务数目	3	6	7	5
测试点是否等分	否	否	否	否

提交源程序文件名

对于 C++ 语言	`city.cpp`	`show.cpp`	`road.cpp`	`sing.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，**不需要建立子文件夹**。
2. 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. 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）。
4. 选手提交的程序源文件必须不大于 100KB。
5. 程序可使用的栈空间内存限制与题目的内存限制一致。
6. **只提供 Linux 格式附加样例文件**。
7. 禁止在源代码中改变编译器参数（如使用 `#pragma` 命令），禁止使用系统结构相关指令（如内联汇编）和其他可能造成不公平的方法。
8. 评测时采用的机器配置为：Windows 11, i7-12700KF @3.60GHz, 32G, Lemonlime。使用的编译器版本为：`x86_64-pc-msys GCC 13.3.0`。上述时限以此配置为准。

城市（city）

【题目描述】

小 W 正处于一座巨大的王国之中，这个王国有 n 座 **城市**，第 i 座城市的编号为 i。为了实现城市与城市之间的通信，每座 **城市** 会通过一些 **基站** 收发信号。为了缓解压力和节约成本，每座城市根据自己的编号二进制表示选择使用的基站。具体来说，编号为 i 的 **城市** 会使用所有编号为 $j \in \text{onbit}(i)$ 的 **基站**，其中 $\text{onbit}(i)$ 表示 i 二进制下为 1 的位（最低位为第 0 位）。例如：5 号城市会使用 0，2 号基站。

而随着时间的流逝，数据传输量越来越大，使得一部分城市之间的通讯越来越不顺畅，于是你决定通过光缆直接连接 **城市**。

对于一个集合 S，定义其 **权值** $\text{val}(S) = \sum_{x \in S} 2^x$。

在两个 **城市** i, j 间连接一条光缆需要花费的代价为 i, j 使用的 **基站** 的并集的权值。同时，为了节约成本，你决定连最少的边。也就是说，建立直接连接的城市形成了一棵 **树**。

现在给出 l, r，请你求出为编号在 $[l, r]$ 内城市建立直接连接的最小花费。也就是说，你需要求出包含 $[l, r]$ 内所有点的 **最小生成树**。

【输入格式】

从文件 ***city.in*** 中读入数据。

输入的第一行包含两个正整数 n, m，分别表示城市个数和询问次数。

接下来 m 行每行两个正整数 l, r，分别表示询问的左右端点。

【输出格式】

输出到文件 ***city.out*** 中。

对于每次询问，输出一行一个整数，表示最小花费。

【样例 1 输入】

```
2 1
1 2
```

【样例 1 输出】

```
3
```

【样例 2 输入】

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

【样例 2 输出】

```
6
9
```

【样例 3】

见选手目录下的 ***city/city3.in*** 与 ***city/city3.ans***。
这个样例满足子任务 1 的约束条件。

【样例 4】

见选手目录下的 ***city/city4.in*** 与 ***city/city4.ans***。
这个样例满足子任务 2 的约束条件。

【样例 5】

见选手目录下的 ***city/city5.in*** 与 ***city/city5.ans***。
这个样例满足子任务 3 的约束条件。

【数据范围】

对于所有测试数据保证：$1 \le n \le 10^9$，$1 \le m \le 10^5$，$1 \le l \le r \le n$。

子任务编号	$n \le$	$m \le$	分值
1	100	10^4	20
2	10^4	10^4	40
3	10^9	10^5	40

烟火（show）

【题目描述】

在成功解决了城市之间的通讯难题之后，小 W 决定举行一场烟火表演。

小 W 对所有城市进行了归类，第 i 座城市的种类为 a_i。

对于一段连续城市区间 $[l,r]$，定义一个种类是 **重要种类** 当且仅当在这段区间内这一种类的城市的数量 >1，一个种类是 **其他种类** 当且仅当在这段区间内这一种类的城市的数量 $=1$。定义一个城市是 **重要城市** 当且仅当该城市的种类是重要种类，一个城市是 **其他城市** 当且仅当该城市的种类是其他种类。

一段区间的城市能进行优美的烟火表演当且仅当重要城市互不相邻，且其他城市也互不相邻。**也就是说，重要城市和其他城市交替出现**。

小 W 现在希望你求出把 n 座城市划分为若干个区间的方案，使得每个区间都能进行 **优美的烟火表演**。由于答案可能很大，你只需要输出答案 $\bmod 325609$。

【输入格式】

从文件 ***show.in*** 中读入数据。

第一行，输入一个正整数 n，表示城市数量。

第二行，输入 n 个非负整数 a_i，表示每个城市的种类。

【输出格式】

输出到文件 ***show.out*** 中。

输出一行一个非负整数，表示方案数取模后的结果。

【样例 1 输入】

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

【样例 1 输出】

```
6
```

【样例 1 解释】

有以下 6 种划分方式：
$\{1\},\{2\},\{1\},\{3\},\{1\}$
$\{1\},\{2\},\{1,3,1\}$
$\{1\},\{2,1,3,1\}$
$\{1,2,1,3,1\}$
$\{1,2,1\},\{3\},\{1\}$
$\{1,2,1,3\},\{1\}$

【样例 2 输入】

```
9
3 2 5 6 9 6 5 2 3
```

【样例 2 输出】

```
4
```

【样例 3】

见选手目录下的 ***show/show3.in*** 与 ***show/show3.ans***。
这个样例满足子任务 1 的约束条件。

【样例 4】

见选手目录下的 ***show/show4.in*** 与 ***show/show4.ans***。
这个样例满足子任务 2 的约束条件。

【样例 5】

见选手目录下的 ***show/show5.in*** 与 ***show/show5.ans***。
这个样例满足子任务 3 的约束条件。

【样例 6】

见选手目录下的 ***show/show6.in*** 与 ***show/show6.ans***。
这个样例满足子任务 4 的约束条件。

【样例 7】

见选手目录下的 ***show/show7.in*** 与 ***show/show7.ans***。
这个样例满足子任务 5 的约束条件。

【样例 8】

见选手目录下的 ***show/show8.in*** 与 ***show/show8.ans***。
这个样例满足子任务 6 的约束条件。

【数据范围】

对于所有测试数据保证：$1 \le n \le 5 \times 10^5$，$1 \le a_i \le n$。

子任务编号	$n \le$	特殊性质	分值
1	300	无	20
2	3000		20
3	10^5		15
4	5×10^5	A	5
5		B	10
6		无	30

特殊性质 A：保证 n 为奇数且 $a_1 = a_3 = ... = a_n = 1$。
特殊性质 B：保证 a_i 在 $[1, n]$ 中随机生成。

道路（road）

【题目描述】

小 W 需要在 m 座城市（编号为 $1 \sim m$）间建造一些公路。由于城市围成了一个环，在编号为 (i, j) 的两座城市间修建公路的代价为 $\min(|i-j|, m-|i-j|)$。

具体的，有 $2 \times n$ 座城市间的公路需要修建，小 W 已经确定了 n 个 A 类城市 $\{a_i\}$ 和 n 个 B 类城市 $\{b_i\}$，每条修建的道路恰好连接一个 A 类城市和一个 B 类城市。此外，每个城市恰好与一条道路相连。

也就是说，**一个合法的修建公路的方案**，构成将 A 类城市视作 **左部点**，B 类城市视作 **右部点** 的 **完全二分图** 的一组 **完美匹配**。

你要做的便是帮小 W 最小化修建公路需要的代价。也就是说，你需要求出这张二分图的 **最小权完美匹配**。

【输入格式】

从文件 ***road.in*** 中读入数据。

本题的测试点包含有多组测试数据。

第一行输入一个正整数 T，表示测试数据组数。

对于每组测试数据：

第一行输入两个正整数 m, n，分别表示城市总数和一类城市的数量。

第二行输入 n 个非负整数 a_i，表示 A 类城市。

第三行输入 n 个非负整数 b_i，表示 B 类城市。

【输出格式】

输出到文件 ***road.out*** 中。

对于每组测试数据，输出一行一个整数，表示最小代价。

【样例 1 输入】

```
1
10 3
1 2 3
4 5 6
```

【样例 1 输出】

```
9
```

【样例 1 解释】

匹配方式为 (a_1, b_1)，(a_2, b_2)，(a_3, b_3) 时，代价为 $3+3+3=9$，容易证明这是最小代价。

【样例 2 输入】

```
1
20 5
1 13 8 15 19
2 6 20 11 7
```

【样例 2 输出】

```
14
```

【样例 3】

见选手目录下的 ***road/road3.in*** 与 ***road/road3.ans***。
这个样例满足子任务 1 的约束条件。

【样例 4】

见选手目录下的 ***road/road4.in*** 与 ***road/road4.ans***。
这个样例满足子任务 2 的约束条件。

【样例 5】

见选手目录下的 ***road/road5.in*** 与 ***road/road5.ans***。
这个样例满足子任务 3 的约束条件。

【样例 6】

见选手目录下的 ***road/road6.in*** 与 ***road/road6.ans***。
这个样例满足子任务 4 的约束条件。

【样例 7】

见选手目录下的 ***road/road7.in*** 与 ***road/road7.ans***。
这个样例满足子任务 5 的约束条件。

【样例 8】

见选手目录下的 ***road/road8.in*** 与 ***road/road8.ans***。
这个样例满足子任务 6 的约束条件。

【样例 9】

见选手目录下的 ***road/road9.in*** 与 ***road/road9.ans***。
这个样例满足子任务 7 的约束条件。

【数据范围】

对于所有数据，保证 $T \le 5 \times 10^5$，$\sum m \le 5 \times 10^7$，$\sum n \le 10^7$，$1 \le a_i, b_i \le m$，每组数据输入的 $2n$ 个城市编号两两不同。

子任务编号	$\sum m \le$	$\sum n \le$	特殊性质	分值
1	10^7	10	无	5
2		20		5
3		1000		10
4		5000		20
5		5×10^5	A	15
6			无	35
7	5×10^7	10^7		10

特殊性质 A：保证 $a_i < b_i$。

本题输入输出量较大，请使用较为快速的输入输出方式。

本题下发快速读入模板，需要的选手可自行取用。请选手注意在调用一次 read 函数之后不得使用其它的输入方式。

具体使用说明见选手目录下的 read.cpp。

合唱（sing）

【题目描述】

顺利完成城市规划的小 W 准备举办一场庆功宴。

庆功宴上，p 个 bot 围着圆桌坐成一圈，并按顺时针顺序依次编号为 $1, 2, ..., p$。

根据计划，他们会进行 q 次合唱，合唱内容为歌手李健的《贝加尔湖畔》。

为了活跃气氛，小 W 会使用一种看似随机的方式决定参与合唱的人选。每次合唱前，小 W 会选择三个整数 x, k, l 并从第 x 个 bot 开始沿顺时针方向 $1, 2, ..., k$ 循环报数。前 l 个报到 k 的 bot 会参加此次合唱。为保证所有合唱都能顺利进行，bot 总数 p 是素数。可以证明当 $1 \leq k < p$ 时，前 p 个报到 k 的 bot 各不相同。

小 T 也打算参加庆功宴。他认为编号为 i 的 bot 唱《贝加尔湖畔》的美妙度为 a_i，而一次合唱的美妙度为所有演唱者的美妙度之和。现在他知道了每次合唱 x, k, l 的值，并希望你算出对应的美妙度，以便最优化听歌体验。

【输入格式】

第一行输入两个正整数 p, q，分别表示 bot 个数和询问次数。

第二行输入 p 个正整数 a_i，表示每个 bot 的美妙度。

接下来 q 行每行三个正整数，依次为每次合唱对应的 x, k, l。

【输入格式】

对于每次询问，输出一行一个整数，表示合唱的美妙度。

【样例 1 输入】

```
17 3
1 1 4 5 1 4 1 8 5 3 7 1 7 9 3 6 3
5 1 3
12 4 11
6 6 6
```

【样例 1 输出】

```
6
42
17
```

【样例 2 输入】

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

【样例 2 输出】

```
12
4
12
8
12
```

【样例 3】

见选手目录下的 ***sing/sing3.in*** 与 ***sing/sing3.ans***。
这个样例满足子任务 1 的约束条件。

【样例 4】

见选手目录下的 ***sing/sing4.in*** 与 ***sing/sing4.ans***。
这个样例满足子任务 2 的约束条件。

【样例 5】

见选手目录下的 ***sing/sing5.in*** 与 ***sing/sing5.ans***。
这个样例满足子任务 3 的约束条件。

【样例 6】

见选手目录下的 ***sing/sing6.in*** 与 ***sing/sing6.ans***。
这个样例满足子任务 4 的约束条件。

【样例 7】

见选手目录下的 ***sing/sing7.in*** 与 ***sing/sing7.ans***。
这个样例满足子任务 5 的约束条件。

【数据范围】

对于所有数据，保证 $1 \le p, q \le 10^5$，$1 \le a_i \le 10^9$，$1 \le x, l \le p$，$1 \le k < p$，且 p 为质数。

子任务编号	$p \le$	$q \le$	特殊性质	分值
1	1000	1000	无	10
2	10^5	10^5	A	10
3			B	10
4			C	20
5			无	50

特殊性质 A：保证 $a_i = i$。
特殊性质 B：保证 $k \le 100$。
特殊性质 C：令 k^{-1} 表示 k 在模 p 意义下的乘法逆元，保证 $k^{-1} \le 100$。