

NOI2024 联合省选

第一试

时间：2024 年 3 月 2 日 08:30 ~ 13:00

| | | | |
|---------|----------|----------|--------------|
| 题目名称 | 季风 | 魔法手杖 | 虫洞 |
| 题目类型 | 传统型 | 传统型 | 传统型 |
| 目录 | wind | xor | wormhole |
| 可执行文件名 | wind | xor | wormhole |
| 输入文件名 | wind.in | xor.in | wormhole.in |
| 输出文件名 | wind.out | xor.out | wormhole.out |
| 每个测试点时限 | 0.5 秒 | 1.5 秒 | 2.0 秒 |
| 内存限制 | 512 MiB | 1024 MiB | 512 MiB |
| 测试点数目 | 10 | 25 | 25 |
| 测试点是否等分 | 是 | 是 | 是 |

提交源程序文件名

| | | | |
|-----------|----------|---------|--------------|
| 对于 C++ 语言 | wind.cpp | xor.cpp | wormhole.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：Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz，内存 16GB。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。

季风 (wind)

【题目背景】

生活在二维平面的小 X 准备拜访小 Y，但由于气候的变化，平面上刮起了季风。小 X 想知道季风的影响下，TA 至少要多少天能够到达小 Y 的家，但小 X 也是第一次遇见这种怪事，所以请精通算法的你来帮忙。

【题目描述】

给定 n, k, x, y 和 $2n$ 个整数 $x_0, y_0, x_1, y_1, \dots, x_{n-1}, y_{n-1}$ 。

找到最小的非负整数 m ，使得存在 $2m$ 个实数 $x'_0, y'_0, x'_1, y'_1, \dots, x'_{m-1}, y'_{m-1}$ 满足以下条件，或报告不存在这样的 m ：

- $\sum_{i=0}^{m-1} (x'_i + x_{i \bmod n}) = x$;
- $\sum_{i=0}^{m-1} (y'_i + y_{i \bmod n}) = y$;
- $\forall 0 \leq i \leq m-1, |x'_i| + |y'_i| \leq k$ 。

特别地， $m = 0$ 时，认为 $(\sum_{i=0}^{m-1} (x'_i + x_{i \bmod n}))$ 和 $(\sum_{i=0}^{m-1} (y'_i + y_{i \bmod n}))$ 均为 0。

【输入格式】

从文件 `wind.in` 中读入数据。

本题有多组测试数据。输入的第一行一个整数 T 表示测试数据组数。

对于每组测试数据，

- 第一行四个整数 n, k, x, y ,
- 接下来 n 行，第 i 行两个整数 x_{i-1}, y_{i-1} 。

【输出格式】

输出到文件 `wind.out` 中。

对于每组测试数据输出一行一个整数，如果存在满足题意的 m ，输出其最小可能值，否则输出 `-1`。

【样例 1 输入】

```
1 4
2 1 2 2 2
3 1 1
4 1 2 -2 -2
5 1 1
6 1 2 0 0
```

```
7 1 1
8 2 1000000000 1000000000 1000000000
9 -999999999 0
10 -1000000000 0
```

【样例 1 输出】

```
1 1
2 -1
3 0
4 3999999999
```

【样例 1 解释】

- 该组样例共有四组测试数据。
- 对于第一组测试数据，取 $m = 1$ ， $(x'_0, y'_0) = (1, 1)$ 满足条件，可以证明不存在更小的 m 满足条件；
 - 对于第二组测试数据，可以证明不存在任何非负整数 m 满足条件；
 - 对于第三组测试数据，取 $m = 0$ 满足条件，可以证明不存在更小的 m 满足条件。

【样例 2】

见选手目录下的 *wind/wind2.in* 与 *wind/wind2.ans*。

该组样例共有八十组测试数据，所有测试数据均满足 $n = 1$ 。其中测试数据 1 ~ 20 满足特殊性质 A，21 ~ 40 满足特殊性质 B，41 ~ 60 满足特殊性质 C。

【样例 3】

见选手目录下的 *wind/wind3.in* 与 *wind/wind3.ans*。

该组样例共有六十组测试数据，所有测试数据均满足 $n = 200$ 。其中测试数据 1 ~ 20 满足特殊性质 A，21 ~ 40 满足特殊性质 B。

【子任务】

- 设 $\sum n$ 为单个测试点内所有测试数据的 n 的和。对于所有测试数据，
- $1 \leq T \leq 5 \times 10^4$ ；
 - $1 \leq n \leq 10^5$ ， $1 \leq \sum n \leq 10^6$ ；
 - $0 \leq |x|, |y|, |x_i|, |y_i|, k \leq 10^8$ 。

| 测试点编号 | $n \leq$ | $\sum n \leq$ | 特殊性质 |
|-------|----------|---------------|------|
| 1 | 1 | 300 | A |
| 2 | | | B |
| 3 | | | C |
| 4 | | | 无 |
| 5 | 200 | 5,000 | A |
| 6 | | | B |
| 7 | | | 无 |
| 8 | 10^4 | 10^5 | A |
| 9 | | | B |
| 10 | 10^5 | 10^6 | 无 |

特殊性质 A: $\forall 0 \leq i \leq n - 1, |x_i| + |y_i| \leq k$ 。

特殊性质 B: $k = 0$ 。

特殊性质 C: $x_0 = y_0 = 0$ 。

【提示】

本题输入文件较大，请使用较为快速的输入方式。

魔法手杖 (xor)

【题目描述】

提示：我们在题目描述的最后提供了一份简要的、形式化描述的题面。

C 城是一座魔力之都，以最高的魔法师水平闻名。对于一名魔法师而言，最重要的固然是魔法手杖和镶嵌在手杖上的魔法水晶。

每个魔法手杖和魔法水晶都可以用**魔力值**来衡量其能力大小，一个魔法手杖的魔力值是镶嵌在其上的所有魔法水晶中魔力值的最小值。

小 ω 是 C 城的一名见习魔法师，他想加强他的魔法手杖。在加强之前，小 ω 的魔法手杖镶嵌着 n 颗魔法水晶，它们的魔力值分别为 a_1, a_2, \dots, a_n 。

小 ω 准备使用一次强力的秘术来加强他的手杖。这一次秘术中，他可以任意选择 x ，然后将所有魔法水晶的魔力值由 a_i 变为 $(a_i \oplus x)$ ，其中 \oplus 表示按位异或。由于小 ω 能力有限， a_1, a_2, \dots, a_n 和 x 都是 $[0, 2^k - 1]$ 中的整数。

小 ω 还发现这个秘术可以定向加强。具体地，他可以花费 b_i 的体力值对第 i 个魔法水晶进行定向加强，将原本应变为 $(a_i \oplus x)$ 的魔力值变为 $(a_i + x)$ 。小 ω 能力有限，因此他定向加强所花费的体力值总和不能超过 m ，且每个水晶只能被定向加强至多一次。

小 ω 想知道他在加强魔法手杖后，魔法手杖的魔力值最大能为多少，但他并不会算，所以请你来帮他计算。

形式化的：给定 a_1, a_2, \dots, a_n 以及 b_1, b_2, \dots, b_n ，满足 $a_i \in [0, 2^k - 1]$ 以及 $b_i \geq 0$ ，你需要给出 $S \subseteq \{1, 2, \dots, n\}$ 以及 $x \in [0, 2^k - 1]$ 满足以下条件：

- $\sum_{i \in S} b_i \leq m$;
 - 满足以上条件的前提下，最大化 $val(S, x) = \min(\min_{i \in S} (a_i + x), \min_{i \in U \setminus S} (a_i \oplus x))$ 的值。
- 你只需要给出最大的 $val(S, x)$ 的值即可。

【输入格式】

从文件 `xor.in` 中读入数据。

本题有多组测试数据。输入的第一行包含两个整数 c, T ，表示测试点编号与测试数据组数。样例中的 c 表示该样例的数据范围与第 c 个测试点的数据范围相同。

接下来依次给出每组输入数据，对于每组数据：

- 第一行三个整数 n, m, k ;
- 第二行 n 个整数 a_1, a_2, \dots, a_n ，分别表示每个魔法水晶的初始魔力值;
- 第三行 n 个整数 b_1, b_2, \dots, b_n ，分别表示每个魔法水晶定向加强需要的体力值。

【输出格式】

输出到文件 `xor.out` 中。

对于每组测试数据输出一行一个整数表示小 ω 能获得魔法手杖魔力值的最大值。

【样例 1 输入】

```
1 1 2
2 5 2 3
3 1 1 2 3 7
4 1 1 0 3 2
5 1 1 1
6 1
7 0
```

【样例 1 输出】

```
1 5
2 2
```

【样例 1 解释】

- 对于第一组数据，一种可行的方案为：定向强化魔法水晶 5（即 $S = \{5\}$ ）并取 $x = 4$ 。最后得到的魔法水晶魔力值分别为 5, 5, 6, 7, 11，故魔法手杖的魔力值为 5。可以证明不存在更优方案。
- 对于第二组数据，一种可行的方案为：定向强化魔法水晶 1（即 $S = \{1\}$ ）并取 $x = 1$ 。

【样例 2】

见选手目录下的 `xor/xor2.in` 与 `xor/xor2.ans`。
该组样例满足 $c = 4$ 。

【样例 3】

见选手目录下的 `xor/xor3.in` 与 `xor/xor3.ans`。
该组样例满足 $c = 7$ 。

【样例 4】

见选手目录下的 `xor/xor4.in` 与 `xor/xor4.ans`。
该组样例满足 $c = 9$ 。

【样例 5】

见选手目录下的 `xor/xor5.in` 与 `xor/xor5.ans`。
该组样例满足 $c = 11$ 。

【样例 6】

见选手目录下的 `xor/xor6.in` 与 `xor/xor6.ans`。
该组样例满足 $c = 14$ 。

【样例 7】

见选手目录下的 `xor/xor7.in` 与 `xor/xor7.ans`。
该组样例满足 $c = 22$ 。

【子任务】

设 $\sum n$ 表示单组测试点各组数据 n 的和。对于所有测试数据，

- $T \geq 1$;
- $1 \leq n \leq 10^5, 1 \leq \sum n \leq 5 \times 10^5$;
- $0 \leq m \leq 10^9$;
- $0 \leq k \leq 120$;
- $\forall 1 \leq i \leq n, 0 \leq a_i < 2^k$;
- $\forall 1 \leq i \leq n, 0 \leq b_i \leq 10^9$ 。

| 测试点编号 | $\sum n \leq$ | $n \leq$ | $m \leq$ | $k \leq$ | 特殊性质 |
|---------|-----------------|----------|----------|----------|------|
| 1 ~ 3 | 10 | 10 | 10^9 | 10 | / |
| 4 ~ 6 | 5×10^5 | 10^5 | 0 | 30 | A |
| 7, 8 | | 2 | 10^9 | | B |
| 9, 10 | | 10^5 | | | |
| 11 ~ 13 | | | | | C |
| 14, 15 | 500 | 10^2 | 10^9 | 60 | / |
| 16 ~ 18 | 5×10^4 | 10^4 | | | |
| 19 ~ 21 | 3×10^5 | 10^5 | | | |
| 22 ~ 25 | 5×10^5 | | | | |

特殊性质 A: $m = 0; \forall 1 \leq i \leq n, b_i \geq 1$ 。
特殊性质 B: $m = 1; \forall 1 \leq i \leq n, b_i \in \{1, 2\}$, 且至多只有 1 个 i 满足 $b_i = 1$ 。
特殊性质 C: $m = 1; \forall 1 \leq i \leq n, b_i \in \{1, 2\}$ 。

【提示】

本题输入文件较大，请使用较为快速的输入方式。

在评测环境中，你可以使用 128 位有符号整型类型 `__int128`，它可以存储范围在 $[-2^{127}, 2^{127} - 1]$ 内的整数，使用方法与其他整型类型基本一致。

需要注意，该类型无法使用诸如 `cin/cout` 或 `scanf/printf` 等常规输入输出方式进行输入输出。我们在选手目录下提供了一份 `__int128` 的输入输出函数实现供选手选择使用。

虫洞 (wormhole)

【题目描述】

E 国有 n 个城市，编号为 1 至 n 。为了让城市之间的来往更加便利，E 国的交通部想在 n 个城市间建造一些虫洞。每条虫洞是一条单向的从某个城市到另一个城市的通道。允许通道的起点和终点是同一个城市，也允许两个城市之间有多个虫洞连接。

为了区分虫洞的建造时间，交通部给每一条虫洞一个正整数的编号。

我们称一种虫洞的建造方案是好的，若它满足如下四个条件：

- (1) 存在一个非负整数 d 使得每个城市恰好是 d 条虫洞的起点，也恰好是 d 条虫洞的终点。
- (2) 对于每个城市而言，在以它为起点的虫洞的编号中，1 到 d 恰好各出现一次。
- (3) 对于每个城市而言，在以它为终点的虫洞的编号中，1 到 d 恰好各出现一次。
- (4) 任意选取一个城市 u 和正整数 $1 \leq j_1, j_2 \leq d$ 。设从 u 出发，先经过一次编号为 j_1 的虫洞，再经过一次编号为 j_2 的虫洞，到达城市 v_1 。设从 u 出发，先经过一次编号为 j_2 的虫洞，再经过一次编号为 j_1 的虫洞，到达城市 v_2 。则条件 $v_1 = v_2$ 必定满足。

特别地，不建造任何虫洞的方案也是好的。

现在，建造师已建造了 mn 条虫洞，且给了它们 $1 \sim m$ 的编号，此时这样的建造方案是好的。他想要新建造 kn 条虫洞，并给它们 $(m+1) \sim (m+k)$ 的编号。他必须保证这 $(m+k)n$ 条虫洞形成的建造方案仍然是好的。他想知道有多少种新建造 kn 条虫洞的方法，使得这 $(m+k)n$ 条虫洞形成的建造方案是好的。

由于答案很大，你只需要求出方案数除以 998244353 的余数。

【输入格式】

从文件 *wormhole.in* 中读入数据。

输入的第一行四个非负整数 c, n, m, k ，其中 c 表示测试点编号。样例中的 c 表示该样例的数据范围与第 c 个测试点的数据范围相同。

接下来 nm 行，每行三个正整数 u, v, w ，表示一条编号为 w 的，起点为 u 号城市，终点为 v 号城市的虫洞。

【输出格式】

输出到文件 *wormhole.out* 中。

输出一行整数，表示方案数除以 998244353 的余数。

【样例 1 输入】

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 4 | 1 | 1 |
| 2 | 1 | 2 | 1 | |
| 3 | 2 | 1 | 1 | |
| 4 | 3 | 4 | 1 | |
| 5 | 4 | 3 | 1 | |

【样例 1 输出】

| | |
|---|---|
| 1 | 8 |
|---|---|

【样例 1 解释】

在该组样例中，已经建造的编号为 1 的虫洞为 $1 \rightarrow 2, 2 \rightarrow 1, 3 \rightarrow 4, 4 \rightarrow 3$ 。为了使 8 条虫洞形成的建造方案是好的，新建造的编号为 2 的虫洞可能有 8 种情形：

- (1) $1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4$
- (2) $1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 4, 4 \rightarrow 3$
- (3) $1 \rightarrow 2, 2 \rightarrow 1, 3 \rightarrow 3, 4 \rightarrow 4$
- (4) $1 \rightarrow 2, 2 \rightarrow 1, 3 \rightarrow 4, 4 \rightarrow 3$
- (5) $1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 1, 4 \rightarrow 2$
- (6) $1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 2, 4 \rightarrow 1$
- (7) $1 \rightarrow 4, 2 \rightarrow 3, 3 \rightarrow 1, 4 \rightarrow 2$
- (8) $1 \rightarrow 4, 2 \rightarrow 3, 3 \rightarrow 2, 4 \rightarrow 1$

【样例 2】

见选手目录下的 *wormhole/wormhole2.in* 与 *wormhole/wormhole2.ans*。
该样例的 $c = 2$ ，它满足第 2 个测试点的限制条件。

【样例 3】

见选手目录下的 *wormhole/wormhole3.in* 与 *wormhole/wormhole3.ans*。
该样例的 $c = 5$ ，它满足第 5 个测试点的限制条件。

【样例 4】

见选手目录下的 *wormhole/wormhole4.in* 与 *wormhole/wormhole4.ans*。
该样例的 $c = 7$ ，它满足第 7 个测试点的限制条件。

【样例 5】

见选手目录下的 *wormhole/wormhole5.in* 与 *wormhole/wormhole5.ans*。
该样例的 $c = 9$ ，它满足第 9 个测试点的限制条件。

【样例 6】

见选手目录下的 *wormhole/wormhole6.in* 与 *wormhole/wormhole6.ans*。
该样例的 $c = 11$ ，它满足第 11 个测试点的限制条件。

【样例 7】

见选手目录下的 *wormhole/wormhole7.in* 与 *wormhole/wormhole7.ans*。
该样例的 $c = 15$ ，它满足第 15 个测试点的限制条件。

【样例 8】

见选手目录下的 *wormhole/wormhole8.in* 与 *wormhole/wormhole8.ans*。
该样例的 $c = 17$ ，它满足第 17 个测试点的限制条件。

【样例 9】

见选手目录下的 *wormhole/wormhole9.in* 与 *wormhole/wormhole9.ans*。
该样例的 $c = 20$ ，它满足第 20 个测试点的限制条件。

【样例 10】

见选手目录下的 *wormhole/wormhole10.in* 与 *wormhole/wormhole10.ans*。
该样例的 $c = 22$ ，它满足第 22 个测试点的限制条件。

【子任务】

对于所有测试点，

- $1 \leq n \leq 2 \cdot 10^3$, $0 \leq m \leq 10^3$, $1 \leq k \leq 10^{15}$;
- $1 \leq u, v \leq n$, $1 \leq w \leq m$;
- 保证初始建造的 mn 条虫洞构成一个好的建造方案。

| 测试点编号 | n | m | k |
|---------|---------------------|-------------|--|
| 1 ~ 4 | ≤ 5 | ≤ 3 | ≤ 3 |
| 5 ~ 6 | $\leq 2 \cdot 10^3$ | $= 0$ | $= 1$ $\leq 10^3$ $\leq 10^{15}$ |
| 7 ~ 8 | $\leq 10^2$ | $= 1$ | |
| 9 ~ 10 | | ≤ 10 | |
| 11 ~ 14 | | $= 0$ | |
| 15 ~ 16 | | ≤ 10 | |
| 17 ~ 19 | | ≤ 10 | |
| 20 ~ 21 | $\leq 2 \cdot 10^3$ | $\leq 10^3$ | $\leq 10^2$ |
| 22 ~ 25 | | | $\leq 10^{15}$ |

【提示】

本题部分测试点输入规模较大，我们推荐你使用较为快速的读入方式。