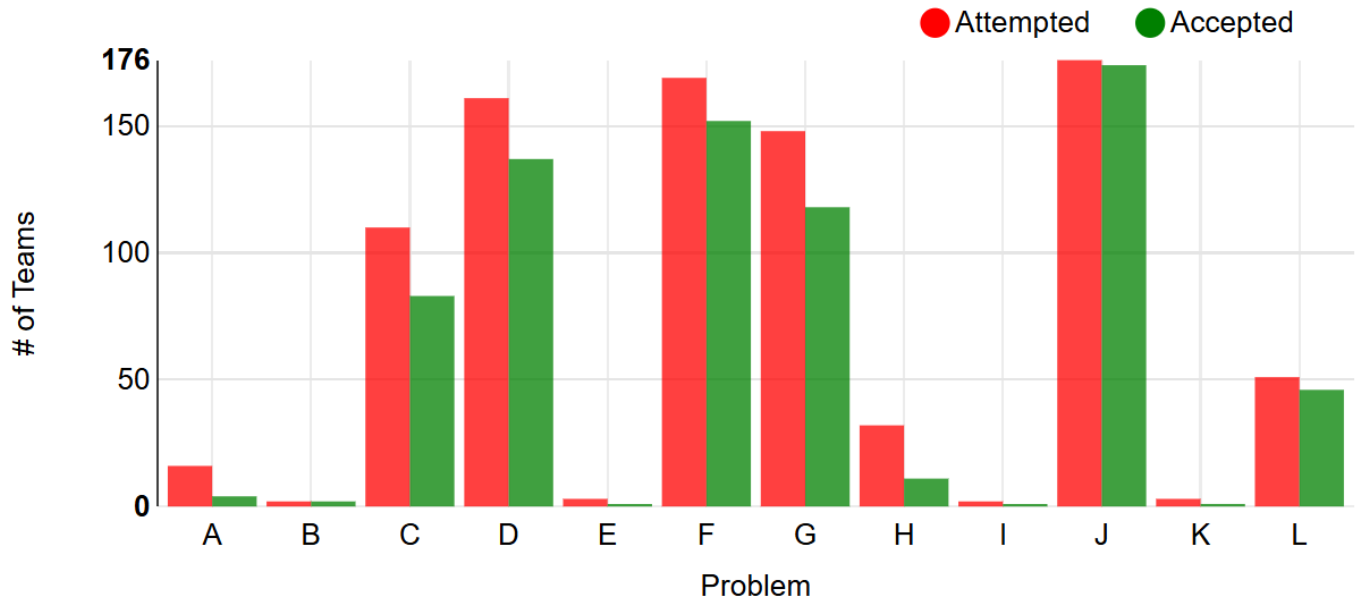
第十八届吉林省省赛简要题解

Nanani Fan Club

May 24, 2025

前言

- 预计难度：JF < CDGL < ABH < EIK。
- 最终难度（封榜时）：



J. Odd-Even Game

简述题意

给定两个数字 a, b，保证其中有一个是奇数、另一个是偶数。若奇数大于偶数输出 1，否则输出 2。

J. Odd-Even Game

简述题意

给定两个数字 a, b，保证其中有一个是奇数、另一个是偶数。若奇数大于偶数输出 1，否则输出 2。

	a 为奇数，b 为偶数	a 为偶数，b 为偶数
$a > b$	1	2
$b > a$	2	1

以你喜欢的方式写判断语句即可。

F - Ever Forever

题目大意

给定一个长度为 n 的字符串，求：

$$\sum_{1 \le i < j \le n} [s_i = \texttt{'e'}] \times [s_j = \texttt{'f'}] \times (j - i)$$

F - Ever Forever

题目大意

给定一个长度为 n 的字符串，求：

$$\sum_{1 \le i < j \le n} [s_i = \texttt{'e'}] \times [s_j = \texttt{'f'}] \times (j - i)$$

- 暴力模拟即可，时间复杂度 $O(n^2)$

F - Ever Forever

题目大意

给定一个长度为 n 的字符串，求：

$$\sum_{1 \le i < j \le n} [s_i = \texttt{'e'}] \times [s_j = \texttt{'f'}] \times (j - i)$$

- 暴力模拟即可，时间复杂度 $O(n^2)$
- 需要注意 *long long* 的使用。

F - Ever Forever

题目大意

给定一个长度为 n 的字符串，求：

$$\sum_{1 \le i < j \le n} [s_i = \texttt{'e'}] \times [s_j = \texttt{'f'}] \times (j - i)$$

- 暴力模拟即可，时间复杂度 $O(n^2)$
- 需要注意 *long long* 的使用。
- 可以通过后缀和轻松优化到 $O(n)$，不过超出了本题签到题的定位。

C. SSPPSPSP

简述题意

给定参数序列 a 以及巨运算符序列 op，计算：

$$\mathrm{power} = [\mathrm{op}_1]_{x_1=0}^{n-1}[\mathrm{op}_2]_{x_2=0}^{n-1}\cdots[\mathrm{op}_k]_{x_k=0}^{n-1}\ a_{(x_1+x_2+\cdots+x_k)\bmod n}$$

C. SSPPSPSP

简述题意

给定参数序列 a 以及巨运算符序列 op，计算：

$$\text{power} = [\text{op}_1]_{x_1=0}^{n-1}[\text{op}_2]_{x_2=0}^{n-1}\cdots[\text{op}_k]_{x_k=0}^{n-1}\, a_{(x_1+x_2+\cdots+x_k)\bmod n}$$

注意到：

- $\sum_{i=0}^{n-1} a_{(x_1+x_2+\cdots+x_k)\bmod n} = \sum_{i=0}^{n-1} a_{(C+x_k)\bmod n} = \sum_{i=0}^{n-1} a$；
- $\prod_{i=0}^{n-1} a_{(x_1+x_2+\cdots+x_k)\bmod n} = \prod_{i=0}^{n-1} a_{(C+x_k)\bmod n} = \prod_{i=0}^{n-1} a$。

C. SSPPSPSP

简述题意

给定参数序列 a 以及巨运算符序列 op，计算：

$$\text{power} = [\text{op}_1]_{x_1=0}^{n-1}[\text{op}_2]_{x_2=0}^{n-1}\cdots[\text{op}_k]_{x_k=0}^{n-1}\, a_{(x_1+x_2+\cdots+x_k)\bmod n}$$

注意到：

- $\sum_{i=0}^{n-1} a_{(x_1+x_2+\cdots+x_k)\bmod n} = \sum_{i=0}^{n-1} a_{(C+x_k)\bmod n} = \sum_{i=0}^{n-1} a$；
- $\prod_{i=0}^{n-1} a_{(x_1+x_2+\cdots+x_k)\bmod n} = \prod_{i=0}^{n-1} a_{(C+x_k)\bmod n} = \prod_{i=0}^{n-1} a$。

即最后一个巨运算符的运算结果总是定值，总可以往前计算，遇到 Σ 就乘上 n，否则做 n 次方，可在 $O(nm)$ 或者 $O(n+m\log n)$ 复杂度求解。

D. Coprime

注意到相邻两个之间质数的最大可能间隔较小。假设 x 和 y 之间最大的质数为 z（如果存在），可知 z 一定与 x 互质，同时由于 $y - z$ 小于最大质数间隔，对较大的 y 一定有 $y < 2z$，因此 z 与 y 互质，z 即为符合要求的答案。

D. Coprime

注意到相邻两个之间质数的最大可能间隔较小。假设 x 和 y 之间最大的质数为 z（如果存在），可知 z 一定与 x 互质，同时由于 $y-z$ 小于最大质数间隔，对较大的 y 一定有 $y<2z$，因此 z 与 y 互质，z 即为符合要求的答案。

因此，只需要从 $y-1$ 开始倒着枚举 z 即可，由于枚举到的第一个质数一定是解，枚举次数上限不会高于最大质数间隔（这是一个很宽的上界，实测根本达不到）。

D. Coprime

在本题数据范围中，最大相邻质数间隔出现在 4302407359 与 $4302407359 + 354$。对于做题而言，不需要知道具体的值是多少，只需要知道相邻质数间隔不会很大即可通过此题。

参阅

关于质数间隔的更多的资料可查阅以下链接：

- https://oeis.org/A002386
- http://primerecords.dk/primegaps/maximal.htm

G. Rock-Paper-Scissors

简述题意

小 V 和小 Λ 玩石头剪刀布，一共有 n 局，其中：

- 小 V 恰好会出 r_1 个石头，s_2 个剪刀，p_3 个布；
- 小 Λ 恰好会出 r_2 个石头，s_2 个剪刀，p_2 个布。

某一局中，若小 V 赢了则总分 $+1$，输了总分 -1，平了不变。问总分的最大值和最小值。

G. Rock-Paper-Scissors

以最大化积分为例，我们可以先贪心地让小 V 尽量赢，即进行：

$$t \leftarrow \min(r_1, s_2), ans \leftarrow ans + t, r_1 \leftarrow r_1 - t, s_2 \leftarrow s_2 - t$$

$$t \leftarrow \min(s_1, p_2), ans \leftarrow ans + t, s_1 \leftarrow s_1 - t, p_2 \leftarrow p_2 - t$$

$$t \leftarrow \min(p_1, r_2), ans \leftarrow ans + t, p_1 \leftarrow p_1 - t, r_2 \leftarrow r_2 - t$$

G. Rock-Paper-Scissors

以最大化积分为例，我们可以先贪心地让小 V 尽量赢，即进行：

$$t \leftarrow \min(r_1, s_2), ans \leftarrow ans + t, r_1 \leftarrow r_1 - t, s_2 \leftarrow s_2 - t$$
$$t \leftarrow \min(s_1, p_2), ans \leftarrow ans + t, s_1 \leftarrow s_1 - t, p_2 \leftarrow p_2 - t$$
$$t \leftarrow \min(p_1, r_2), ans \leftarrow ans + t, p_1 \leftarrow p_1 - t, r_2 \leftarrow r_2 - t$$

在这之后，小 V 已经只能平局或失败，为了最优他只能尽可能平局，即进行：

$$t \leftarrow \min(r_1, r_2), r_1 \leftarrow r_1 - t, r_2 \leftarrow r_2 - t$$
$$t \leftarrow \min(s_1, s_2), s_1 \leftarrow s_1 - t, s_2 \leftarrow s_2 - t$$
$$t \leftarrow \min(p_1, p_2), p_1 \leftarrow p_1 - t, p_2 \leftarrow p_2 - t$$

G. Rock-Paper-Scissors

当然，进行完之后可能也会有小 V 不得不输的情况（例如小 V 只有一张剪刀，而小 Λ 只有一张石头），所以需要再进行：

$$t \leftarrow \min(s_1, r_2), ans \leftarrow ans - t, s_1 \leftarrow s_1 - t, r_2 \leftarrow r_2 - t$$

$$t \leftarrow \min(p_1, s_2), ans \leftarrow ans - t, p_1 \leftarrow p_1 - t, s_2 \leftarrow s_2 - t$$

$$t \leftarrow \min(r_1, p_2), ans \leftarrow ans - t, r_1 \leftarrow r_1 - t, p_2 \leftarrow p_2 - t$$

最终 ans 的值即为最大化积分。

最小化积分只需要将上述两个过程顺序反过来进行即可，总时间复杂度 $O(T)$。

L. Good Matrix

简述题意

给定 n, m。我们认为一个 $n \times m$ 大小的 01 矩阵 A 是好的，当且仅当：

$$A_{ij} = \left(\bigoplus_{k=1}^{m} A_{ik} \right) \oplus \left(\bigoplus_{k=1}^{n} A_{kj} \right)$$

计算其方案数。

L. Good Matrix

简述题意

给定 n, m。我们认为一个 $n \times m$ 大小的 01 矩阵 A 是好的，当且仅当：

$$A_{ij} = \left(\bigoplus_{k=1}^{m} A_{ik} \right) \oplus \left(\bigoplus_{k=1}^{n} A_{kj} \right)$$

计算其方案数。

- 设 r_i 为第 i 行元素的异或和，即 $\bigoplus_k A_{ik}$；
- 设 c_j 为第 j 行元素的异或和，即 $\bigoplus_k A_{kj}$；
- 设 R 为 r_i 的异或和，C 为 c_j 的异或和。

L. Good Matrix

首先有 $A_{ij} = r_i \oplus c_j$，只要确定 $[r_i], [c_j]$ 即可固定整个矩阵。此外，

$$r_i = \bigoplus_{k=1}^{m} A_{ik} = C \oplus ([m \bmod 2 = 1] \times r_i)$$

$$c_j = \bigoplus_{k=1}^{n} A_{kj} = R \oplus ([n \bmod 2 = 1] \times c_j)$$

考虑按照 n, m 的奇偶性分类讨论。

L. Good Matrix

- 当 n, m 均为奇数时，$R = C = 0$，共有 2^{n+m-2} 种填法；

L. Good Matrix

- 当 n, m 均为奇数时，$R = C = 0$，共有 2^{n+m-2} 种填法；
- 当 n, m 均为偶数时，$r_1 = r_2 = \cdots = C, c_1 = c_2 = \cdots = R$，又因为 n, m 均为偶数从而 $C = R = 0$，共有 1 种填法；

L. Good Matrix

- 当 n, m 均为奇数时，$R = C = 0$，共有 2^{n+m-2} 种填法；
- 当 n, m 均为偶数时，$r_1 = r_2 = \cdots = C, c_1 = c_2 = \cdots = R$，又因为 n, m 均为偶数从而 $C = R = 0$，共有 1 种填法；
- 当 n, m 一奇一偶，不妨设 n 为奇数 m 为偶数，则 $R = 0, r_1 = r_2 = \cdots = C$，从而 $0 = R = r_1 \oplus r_2 \oplus \cdots = C$，共有 2^{m-1} 种填法；同理，对于 n 为偶数 m 为奇数，共有 2^{n-1} 种填法。

A. Genius Cirno's Genius Computer

简述题意

有一台超级计算机支持 int1024 的计算，有 8 个寄存器，可对其进行加减乘除以及比较操作。要求在 6666 次操作内求出放在寄存器 a, b, c, d 里的 a_0, b_0, c_0, d_0 对应的分数 a_0/b_0 和 c_0/d_0 的大小关系。

A. Genius Cirno's Genius Computer

题目要求运算过程中不能发生溢出，且 a_0, b_0, c_0, d_0 的值可能很大，因此留给我们操作的空间并不多。

A. Genius Cirno's Genius Computer

题目要求运算过程中不能发生溢出，且 a_0, b_0, c_0, d_0 的值可能很大，因此留给我们操作的空间并不多。

一个简单的想法是，假设 $\lfloor a/b \rfloor$ 和 $\lfloor c/d \rfloor$ 的值（即我们只看这些分数的整数部分）不同，那么可以立即得到大小关系；否则，总是可以令 $a \leftarrow a - \lfloor a/b \rfloor \cdot b$ 以及 $c \leftarrow c - \lfloor c/d \rfloor \cdot d$，这样不会改变两者的大小关系。

A. Genius Cirno's Genius Computer

题目要求运算过程中不能发生溢出，且 a_0, b_0, c_0, d_0 的值可能很大，因此留给我们操作的空间并不多。

一个简单的想法是，假设 $\lfloor a/b \rfloor$ 和 $\lfloor c/d \rfloor$ 的值（即我们只看这些分数的整数部分）不同，那么可以立即得到大小关系；否则，总是可以令 $a \leftarrow a - \lfloor a/b \rfloor \cdot b$ 以及 $c \leftarrow c - \lfloor c/d \rfloor \cdot d$，这样不会改变两者的大小关系。

- 如果操作后 a 或者 c 变成了 0，容易判定原先两个分数的大小关系；
- 否则，两个分数均变为了真分数（分母大于分子），直接比较比较困难，但是可以发现只需要比较两者的倒数 b/a 和 d/c 的大小关系，取反后即为原先两个分数的大小关系。可以递归求解。

A. Genius Cirno's Genius Computer

一个可能的实现如下：

1. $r_0 \leftarrow a \div b$；
2. $r_1 \leftarrow c \div d$；
3. 比较 r_0 和 r_1，若不同则立即结束；
4. $r_0 \leftarrow b \times r_0$；
5. $a \leftarrow a - r_0$；
6. $r_1 \leftarrow d \times r_1$；
7. $c \leftarrow c - r_1$；
8. 比较 a 和 r_2（恒为 0）；
9. 比较 b 和 r_2（恒为 0）。

A. Genius Cirno's Genius Computer

一个可能的实现如下：

1. $r_0 \leftarrow a \div b$；
2. $r_1 \leftarrow c \div d$；
3. 比较 r_0 和 r_1，若不同则立即结束；
4. $r_0 \leftarrow b \times r_0$；
5. $a \leftarrow a - r_0$；
6. $r_1 \leftarrow d \times r_1$；
7. $c \leftarrow c - r_1$；
8. 比较 a 和 r_2（恒为 0）；
9. 比较 b 和 r_2（恒为 0）。

如果 $8, 9$ 步有数字为 0 则立即结束，否则交换 a, b 和 c, d 回到步骤 1 迭代进行。事实上，我们不需要显式地交换寄存器的值，只要交换两者的名字即可。

A. Genius Cirno's Genius Computer

考虑交互次数。这其实是一个辗转相除的过程。

A. Genius Cirno's Genius Computer

考虑交互次数。这其实是一个辗转相除的过程。

最坏情况下 a 和 b、c 和 d 呈现斐波那契数列的形式。斐波那契数列的增长速度关于二进制位数 N 的关系大约为 $N \times \ln 2 / \ln((1+\sqrt{5})/2)$，对于本题 $N = 1023$ 得到大约 $T = 1473$ 轮操作次数，再乘上每轮 9 次操作得到 13261 是不够的。

A. Genius Cirno's Genius Computer

考虑交互次数。这其实是一个辗转相除的过程。

最坏情况下 a 和 b、c 和 d 呈现斐波那契数列的形式。斐波那契数列的增长速度关于二进制位数 N 的关系大约为 $N \times \ln 2 / \ln((1+\sqrt{5})/2)$，对于本题 $N = 1023$ 得到大约 $T = 1473$ 轮操作次数，再乘上每轮 9 次操作得到 13261 是不够的。

然而，在迭代 $T/2$ 次后 a, b, c, d 的值均不超过 $\sqrt{2^{1023}}$，此时直接计算 $a \times c$ 和 $b \times d$ 不会溢出，直接比较即可。

A. Genius Cirno's Genius Computer

Bonus

出题人在出题过程中考虑了一些别的做法，尽管其中一些在后续加强过程中被卡了，但是选手可以参考其过程：

A. Genius Cirno's Genius Computer

Bonus

出题人在出题过程中考虑了一些别的做法，尽管其中一些在后续加强过程中被卡了，但是选手可以参考其过程：

- 尝试使用二分方法套出 a_0, b_0, c_0, d_0 的值。先构造 $r_0 = 2^{1022}$。接着每次令 $r_1 \leftarrow a - r_0$，比较 r_1 和 0 选择用 a 还是 r_1 作为新的 a 进行迭代（b, c, d 同理），最后 $r_0 \leftarrow r_0 \div 2$。出题人使用此思路可以得到 $9N + \varepsilon$ 次的交互过程。

A. Genius Cirno's Genius Computer

Bonus

出题人在出题过程中考虑了一些别的做法，尽管其中一些在后续加强过程中被卡了，但是选手可以参考其过程：

- 尝试使用二分方法套出 a_0, b_0, c_0, d_0 的值。先构造 $r_0 = 2^{1022}$。接着每次令 $r_1 \leftarrow a - r_0$，比较 r_1 和 0 选择用 a 还是 r_1 作为新的 a 进行迭代（b, c, d 同理），最后 $r_0 \leftarrow r_0 \div 2$。出题人使用此思路可以得到 $9N + \varepsilon$ 次的交互过程。
- 尝试使用拆位的思想，设 $a_0 = \mathrm{hi} \times 2^{512} + \mathrm{lo}$ 等，那么计算 $a_0 \times c_0$ 以及 $b_0 \times d_0$ 则能拆成若干个不大的数字相乘。然而此方法受限于有限的寄存器个数以及紧凑的类型范围，实现起来比较复杂。
  验题人通过该思路可以得到 $\mathrm{O}(1) + \mathrm{O}(\log \log N)$ 复杂度的做法，大约使用了 180 次交互次数。

B. Triangle Uika

简述题意

有三名角色 A, B, C 以单位速率行走在二维平面上，每个人的轨迹都是一条折线。需要求出行动过程中三角形 ABC 面积的最大值。

B. Triangle Uika

首先可以将时间轴分为 $O(n)$ 段，每一段中三个人沿着线段匀速行走。

B. Triangle Uika

首先可以将时间轴分为 $O(n)$ 段，每一段中三个人沿着线段匀速行走。

可以固定其中一名角色（例如 A）为参考系，考察其他两名角色（B，C）的运动。运动可视为从起点 $\mathbf{s}_B$（$\mathbf{s}_C$）出发沿着 $\mathbf{v}_B$（$\mathbf{v}_C$）方向的匀速运动。坐标分别为 $\mathbf{s}_B + t\mathbf{v}_B$ 和 $\mathbf{s}_C + t\mathbf{v}_C$。

B. Triangle Uika

首先可以将时间轴分为 $O(n)$ 段，每一段中三个人沿着线段匀速行走。

可以固定其中一名角色（例如 A）为参考系，考察其他两名角色（B，C）的运动。运动可视为从起点 $\mathbf{s}_B$（$\mathbf{s}_C$）出发沿着 $\mathbf{v}_B$（$\mathbf{v}_C$）方向的匀速运动。坐标分别为 $\mathbf{s}_B + t\mathbf{v}_B$ 和 $\mathbf{s}_C + t\mathbf{v}_C$。

三角形的面积可以表示为 $|(\mathbf{s}_B + t\mathbf{v}_B) \times (\mathbf{s}_C + t\mathbf{v}_C)|/2$（叉积）。展开后是关于 t 的二次函数（可能会退化）外面套一层绝对值，形如 $|At^2 + Bt + C|$。

B. Triangle Uika

首先可以将时间轴分为 $O(n)$ 段，每一段中三个人沿着线段匀速行走。

可以固定其中一名角色（例如 A）为参考系，考察其他两名角色（B，C）的运动。运动可视为从起点 $\mathbf{s}_B$（$\mathbf{s}_C$）出发沿着 $\mathbf{v}_B$（$\mathbf{v}_C$）方向的匀速运动。坐标分别为 $\mathbf{s}_B + t\mathbf{v}_B$ 和 $\mathbf{s}_C + t\mathbf{v}_C$。

三角形的面积可以表示为 $|(\mathbf{s}_B + t\mathbf{v}_B) \times (\mathbf{s}_C + t\mathbf{v}_C)|/2$（叉积）。展开后是关于 t 的二次函数（可能会退化）外面套一层绝对值，形如 $|At^2 + Bt + C|$。

有两种做法：

- 该函数的极值只有可能会在端点处或者对称轴处取得，取这三个位置计算并取最大值。需要判断对称轴是否在时间范围内。此方法可能会遇到精度问题，但本题数据未出现；

B. Triangle Uika

首先可以将时间轴分为 $O(n)$ 段，每一段中三个人沿着线段匀速行走。

可以固定其中一名角色（例如 A）为参考系，考察其他两名角色（B，C）的运动。运动可视为从起点 $\mathbf{s}_B$（$\mathbf{s}_C$）出发沿着 $\mathbf{v}_B$（$\mathbf{v}_C$）方向的匀速运动。坐标分别为 $\mathbf{s}_B + t\mathbf{v}_B$ 和 $\mathbf{s}_C + t\mathbf{v}_C$。

三角形的面积可以表示为 $|(\mathbf{s}_B + t\mathbf{v}_B) \times (\mathbf{s}_C + t\mathbf{v}_C)|/2$（叉积）。展开后是关于 t 的二次函数（可能会退化）外面套一层绝对值，形如 $|At^2 + Bt + C|$。

有两种做法：

- 该函数的极值只有可能会在端点处或者对称轴处取得，取这三个位置计算并取最大值。需要判断对称轴是否在时间范围内。此方法可能会遇到精度问题，但本题数据未出现；
- 该函数除去绝对值是凸的，可以使用三分法求解。

H. Another Palindromes Problem

简述题意

给定字符串 s，每次询问其子串 $s[l,r]$ 能否通过 “交换操作” 变为回文串。其中 “交换操作” 的定义如下：

- 对于长度为 n 的字符串 t，选择下标 $1 \le i \le n-2$，交换 t_i 和 t_{i+2}。

H. Another Palindromes Problem

注意到交换只能对同奇偶的位置执行，于是通过有限次交换，我们能（且只能）够将询问区间内的奇位置的子序列、偶位置的子序列分别任意排序。又有询问区间长度为偶数，于是“交换”后回文串的对应关系一定为一个奇数位置对应一个偶数位置。

H. Another Palindromes Problem

注意到交换只能对同奇偶的位置执行，于是通过有限次交换，我们能（且只能）够将询问区间内的奇位置的子序列、偶位置的子序列分别任意排序。又有询问区间长度为偶数，于是“交换”后回文串的对应关系一定为一个奇数位置对应一个偶数位置。

综上，判断一个偶区间能够“交换”为回文的充分必要条件为：区间的奇数位的元素所构成的多重集和偶数位置的元素所构成的多重集相同。

H. Another Palindromes Problem

考虑用哈希维护这个条件，具体来说，我们固定一个底数 B，维护多重集中所有元素 B^{a_i} 的和，在不出现哈希冲突的情况下，两个多重集相等当且仅当他们的哈希值相等。

H. Another Palindromes Problem

考虑用哈希维护这个条件，具体来说，我们固定一个底数 B，维护多重集中所有元素 B^{a_i} 的和，在不出现哈希冲突的情况下，两个多重集相等当且仅当他们的哈希值相等。

我们用线段树维护整个序列**奇数位置以及偶数位置**的 B^{a_i} 的和，题目的区间加操作即转化为线段树上的区间乘操作；若我们将奇数位置的值令为正，偶数位置的值令为负，则区间询问操作即在线段树上求区间和，判断是否为零。

时间复杂度 $O(n + q \log n)$。

E - Eternal Feather

题目大意

给定递推式 $f(i)$：

$$\begin{cases} f(0) = 0, \\ f(1) = 1, \\ f(i) = p \cdot f(i-1) + q \cdot f(i-2) \quad (i \geq 2) \end{cases}$$

求

$$\sum_{a=1}^{n} \sum_{b=1}^{n} \sum_{c=1}^{m} \gcd(f(c^a + 1), f(c^b + 1))$$

E - Eternal Feather

Theorem (引理 1)

$$f(\gcd(n,m)) = \gcd(f(n), f(m))$$

E - Eternal Feather

Theorem (引理 1)

$$f(\gcd(n,m)) = \gcd(f(n), f(m))$$

Theorem (引理 2)

$$\gcd(a^n+1, a^m+1) = \begin{cases} a^{\gcd(n,m)}+1 & \text{如果 } \frac{n}{\gcd(n,m)} \text{ 和 } \frac{m}{\gcd(n,m)} \text{ 均为奇数,} \\ 1 & \text{否则若 } a \text{ 为偶数,} \\ 2 & \text{否则若 } a \text{ 为奇数.} \end{cases}$$

证明均可用数学归纳法 + 无穷递降，此处从略。

E - Eternal Feather

枚举 $d=\gcd(a,b)$，设 $a=a_0d,b=b_0d$。我们需要统计 $a_0\perp b_0$ 且 a_0,b_0 均为奇数的方案数。事实上，该方案数为：

$$\sum_{i=1}^{2i-1\le n/d}\varphi(2i-1)=\varphi(1)+\varphi(3)+\varphi(5)+\cdots$$

E - Eternal Feather

枚举 $d=\gcd(a,b)$，设 $a=a_0d,b=b_0d$。我们需要统计 $a_0\perp b_0$ 且 a_0,b_0 均为奇数的方案数。事实上，该方案数为：

$$\sum_{i=1}^{2i-1\le n/d}\varphi(2i-1)=\varphi(1)+\varphi(3)+\varphi(5)+\cdots$$

对于一个大于 1 的奇数 m，考虑与 $2m$ 互质的数 x（它一定与 m 也互质）：若 x 与 $2m$ 互质，则 $2m-x$ 也与 $2m$ 互质，而 x 和 $2m-x$ 必有一个不超过 m。再因为 $\varphi(2m)=\varphi(2)\varphi(m)=\varphi(m)$，可知上式。

E - Eternal Feather

枚举 $d=\gcd(a,b)$，设 $a=a_0d,b=b_0d$。我们需要统计 $a_0\perp b_0$ 且 a_0,b_0 均为奇数的方案数。事实上，该方案数为：

$$\sum_{i=1}^{2i-1\le n/d}\varphi(2i-1)=\varphi(1)+\varphi(3)+\varphi(5)+\cdots$$

对于一个大于 1 的奇数 m，考虑与 $2m$ 互质的数 x（它一定与 m 也互质）：若 x 与 $2m$ 互质，则 $2m-x$ 也与 $2m$ 互质，而 x 和 $2m-x$ 必有一个不超过 m。再因为 $\varphi(2m)=\varphi(2)\varphi(m)=\varphi(m)$，可知上式。

此外我们可以计算 a_0,b_0 不同时为奇数的方案数为 $2\sum\varphi(i)-1$ 减去上式。

E - Eternal Feather

从而我们能够计算出两类 a_0, b_0 的方案数，可以直接带入式子计算。

我们计算的 $f(a^d + 1)$ 较大，即使是使用矩阵乘法也难以快速求解。注意到对于 f 而言存在不超过 mod^2 大小的循环节（每一项之和前两项的和有关，可以使用鸽巢原理），本题模数为 1225，因而可以快速计算。

E - Eternal Feather

从而我们能够计算出两类 a_0, b_0 的方案数，可以直接带入式子计算。

我们计算的 $f(a^d+1)$ 较大，即使是使用矩阵乘法也难以快速求解。注意到对于 f 而言存在不超过 mod^2 大小的循环节（每一项之和前两项的和有关，可以使用鸽巢原理），本题模数为 1225，因而可以快速计算。

Bonus

本题可以使用扩展欧拉定理 + 杜教筛将复杂度降至 $O(mn^{2/3})$。

- E、F 两题背景均出自 galgame 「*ef - a fairy tale of the two.*」，感兴趣的同学，欢迎与出题人 (bangumi ID: SkyRainWind) 交流。
- Merry Christmas, Yuuko.（模数来源）

I. Black and White Coloring

简述题意

给定 n 个点 m 条边的简单无向图，每个点度数不超过 3，每个点初始没有颜色。

定义图的得分为图中两个端点颜色不同的边的数量。询问有多少种方案将一些点染成黑色或者白色，使得不存在方案将剩余点染色后图的得分都不超过 $\frac{m}{2}$。

I. Black and White Coloring

Theorem (引理 1)

无论小 V 如何染色，小 Λ 都存在一种染色方法，使得最后图的得分不小于 $\frac{m}{2}$。

Proof.

对于小 Λ，考虑让他按照 $1/2$ 相同的概率随机给剩余的点染色，并假设 X 表示图的得分，m_1 为小 V 染色对连接两点都染了色的边，$m_2 = m - m_1$，则有

$$E(X) = m_1 + \frac{m_2}{2} \geq \frac{m}{2}$$

对于任意概率分布，显然存在 $x \geq E(X)$ 使得 $P(X = x) > 0$，故得证。 □

I. Black and White Coloring

综上，小 V 的染色方案应该满足，**无论小 Λ 如何染色，最后图的得分都为 $\frac{m}{2}$**。

假设小 V 染色了的点集为 S，黑色的点集为 S_1，白色的点集为 S_2（$S = S_1 \cup S_2, S_1 \cap S_2 = \varnothing$），未染色的点集为 T。

Theorem (引理 2)

不存在 $\{xy\} \in E$，使得 $x \in T, y \in T$。

Proof.

假设固定其他点的颜色，考虑 x 和 y 的四种不同染色方法，易证一定有两种染色方法得分不同。故矛盾。 □

I. Black and White Coloring

Theorem (引理 3)

不存在 $\{xy\} \in E$，使得 $x \in S, y \in S$。

Proof.

此时 $m_1 > 0$，$E(x) > \frac{m}{2}$，故矛盾。 □

由定理 2,3 可见，S 和 T 构成了二分图的左部和右部。

I. Black and White Coloring

Theorem

任意 $x \in T$，$deg_x = 0$ 或 $deg_x = 2$。且 $deg_x = 2$ 时，其连接的两点 yz 满足 $y \in S_1, z \in S_2$。

Proof.

假设固定除了 x 以外所有点的颜色，如果不满足，x 染不同颜色的得分不同，矛盾。（实际上满足这条只需要 deg_x 是偶数，这也是 $deg_x < 3$ 条件的用处）□

因此，我们可以考虑对于每个 $x \in T, deg_x > 0$，如果这样的 x 都满足 $deg_x = 2$，则建立一条 y 到 z 之间的双向边。我们会得到一个新的图 $G' = (S, E')$，合法的方案则应该是对 G' 的一个合法黑白染色。

显然，对于一个图来说，如果不是二分图，则合法黑白染色的数量为 0。否则应该是 $2^{\text{连通块数量}}$。

综上，我们可以得到本题完整的解法：

- 首先对于每个连通块独立考虑。如果该连通块不是二分图，答案为 0，返回。

显然，对于一个图来说，如果不是二分图，则合法黑白染色的数量为 0。否则应该是 $2^{\text{连通块数量}}$。

综上，我们可以得到本题完整的解法：

- 首先对于每个连通块独立考虑。如果该连通块不是二分图，答案为 0，返回。
- 否则，考虑分别枚举两部为 S 或 T，并将两种情况的方案数相加。

显然，对于一个图来说，如果不是二分图，则合法黑白染色的数量为 0。否则应该是 $2^{\text{连通块数量}}$。

综上，我们可以得到本题完整的解法：

- 首先对于每个连通块独立考虑。如果该连通块不是二分图，答案为 0，返回。
- 否则，考虑分别枚举两部为 S 或 T，并将两种情况的方案数相加。
- 对于其中一种情况，我们对 S 按照上述方法建立新图。如果不是二分图，则返回 0，否则返回 $2^{\text{连通块数量}}$。

显然，对于一个图来说，如果不是二分图，则合法黑白染色的数量为 0。否则应该是 $2^{\text{连通块数量}}$。

综上，我们可以得到本题完整的解法：

- 首先对于每个连通块独立考虑。如果该连通块不是二分图，答案为 0，返回。
- 否则，考虑分别枚举两部为 S 或 T，并将两种情况的方案数相加。
- 对于其中一种情况，我们对 S 按照上述方法建立新图。如果不是二分图，则返回 0，否则返回 $2^{\text{连通块数量}}$。
- 最终答案即为所有连通块的答案之积。

实现只需要两次黑白染色即可，用 dfs 或并查集即可搞定，时间复杂度 $O(n+m)$。

K. Maximum Profit

简述题意

公司共有 n 个员工以及 n 个岗位。第 i 个员工的能力值为 a_i，第 i 个岗位的要求值为 b_i，第 i 个员工就职第 j 个岗位产生的基础收益为 $a_i + b_j + (a_i \oplus b_j)$。

特定的员工就职特定的岗位会产生额外的效益。有 m 条信息 (x, y, w)，第 x 个员工就职第 y 个岗位会产生 w 的额外收益。

给定参数 K，对于 $1 \le k \le K$，求出若恰好有 k 个员工进行就职，产生的总收益的最大值。

K. Maximum Profit

Theorem

$$a_i + b_j + (a_i \oplus b_j) = 2(a_i \mid b_j)$$

注意到二分图带权匹配在增广时不会退流，即源汇边一定不退流。

设当前源汇边已经流过流量连接的点为关键点。可以发现一条点数大于 2 的增广路（不算 s, t）除了开头结尾剩下的点都是关键点。

于是只需要对关键点跑 floyd，然后两端的关键点找非关键点的最大匹配。

K. Maximum Profit

有两种处理方式：

- 一个是直接跑值域的暴力，另一个是写根号分治。由于本题 $V \le 2^{12}$，所以跑暴力可以 AC，当然更优秀的做法就是写根号分治。
- 对于非关键点与非关键点的匹配可以 FWT，关键点找非关键点就写根号分治，所以应该出到 2^{16}，但是因为常数过大没出。

时间复杂度 $O((n+m)K + K^4 + K^2\sqrt{V} + n\sqrt{V} + KV\log V)$，其中 V 为值域，实现时注意常数。

K. Maximum Profit

Bonus

上面的模拟费用流做法是出题人期望的做法。

实际上由于边权的形式非常优美，可以重新在点权的基础进行重建图跑费用流，而且减出来的图是 DAG 具有良好的性质，使用 Dijkstra 费用流科技优化也可通过此题。