杂题选讲

陈东武

广州大学附属中学

2022 年 2 月 24 日

给定两个 `AB?` 字符串 c, d 和正整数 n，求对所有将 `?` 替换为 `A` 或 `B` 的方案，满足 $|S|, |T| \le n$ 且「将 c, d 的 `A` 和 `B` 分别替换为 S 和 T 得到的字符串相等」的 `01` 字符串对 (S, T) 的个数之和 $\bmod (10^9 + 7)$。[1]

$|c|, |d|, n \le 3 \cdot 10^5$

[1] `https://codeforces.com/contest/794/problem/G`

引理

若 $c \neq d$ 则 S 和 T 有长为 $\gcd(|S|, |T|)$ 的整周期。

引理

若 $c \neq d$ 则 S 和 T 有长为 $\gcd(|S|,|T|)$ 的整周期。

证明.

对 $|S|+|T|$ 归纳，当 $|S|=|T|=1$ 时显然成立。

否则不妨设 $|S| \leq |T|$，若 $|S|=|T|$ 则显然 $S=T$，若 $|S|<|T|$ 则把 c,d 的 lcp 去掉，不妨设 $c_1=\mathtt{A}$，$d_1=\mathtt{B}$，则 S 是 T 的前缀，设 $T=S+T'$，并将 c,d 中的 $\mathtt{B}$ 替换为 $\mathtt{AB}$，此时 $|S|+|T'|$ 更小了，且 $c_2 \neq d_2$ 所以 $c \neq d$，由归纳假设知 S,T' 都有长为 $\gcd(|S|,|T'|)$ 的整周期。 □

设 a 是 c 的 A 个数减去 d 的 A 个数，b 是 d 的 B 个数减去 c 的 B 个数，则再加上 $a\cdot|S|=b\cdot|T|$ 即为充要条件。

先特判掉 $c=d$ 时方案数为 $(\sum_{i=1}^{n}2^i)^2$，然后当 $a=b=0$ 时

$$\sum_{i=1}^{n}\sum_{j=1}^{n}2^{\gcd(i,j)}=\sum_{p=1}^{n}2^p\sum_{d=1}^{\lfloor n/p\rfloor}\mu(d)\lfloor\frac{n}{pd}\rfloor^2$$

后面的和式只与 $\lfloor n/p\rfloor$ 有关，整除分块即可 $O(n)$。

否则只有 $ab>0$ 时才有解，方案数为 $\sum_{i=1}^{r}2^i$，其中 $r=\lfloor\frac{n\gcd(a,b)}{\max(a,b)}\rfloor$。

然后要考虑有问号的情况：设 x, y 表示 c, d 的 ? 数量，$f_{a,b}$ 表示上述答案的值，则

$$\begin{aligned} Ans &= \sum_i \sum_j f_{a+i-j,b+(y-j)-(x-i)} \binom{x}{i} \binom{y}{j} \\ &= \sum_{k=-y}^{x} f_{a+k,b+y-x+k} \sum_j \binom{x}{k+j} \binom{y}{j} \\ &= \sum_{k=0}^{x+y} f_{a-y+k,b-x+k} \binom{x+y}{k} \end{aligned}$$

时间复杂度 $O(n+|c|+|d|)$。

给定长为 n 的自然数序列 $a_1, \cdots, a_n$，每次操作选择满足「对所有 $j \neq i$ 都有 $a_j > 0$」的下标 i，令 a_i 加上 n 然后全局减 1，求能得到的序列数量 $\bmod 998244353$。[2]

$n \leq 10^5$，$a_i \leq 10^6$

[2] https://codeforces.com/contest/1188/problem/E

引理 (充要条件)

设 x_i 表示对下标 i 的操作次数，$s=\sum x_i$，则操作可行当且仅当

- 对所有 i 都有 $x_i \geq \lceil\frac{s-a_i}{n}\rceil$；
- 对所有 $t\in[0,s)$ 都有 $t \geq \sum_{i=1}^{n}\lceil\frac{\max(t-a_i,0)}{n}\rceil$。

而两种 $\{x_i\}$ 得到的序列相同当且仅当每一位相差相同，所以数满足 $\min x_i=0$ 的 $\{x_i\}$ 即可。

此时有 $s\leq\max a_i$，所以从小到大枚举 s，不满足引理的第二个条件就直接退出，然后就是插板法。

时间复杂度 $O(\mathrm{Sort}(n)+\max a_i)$。

m 个人参加一场 n 道题的考试，每题一分，已知所有人总成绩为 t，第 i 道题至少有 l_i 人答对，至多有 r_i 人答对。

q 个条件形如第 p_i 名得分为 s_i，求最多能有多少人并列第一，且在这种情况下第一名的分数最大值。需判断无解。[3]

$n, m \leq 10^5$

[3] https://codeforces.com/contest/1305/problem/H

考虑若已知每个人的得分 $s_1 \geq \cdots \geq s_m$ 如何判断可行性，这是上下界可行流的模型：u_i 表示第 i 道题，v_j 表示第 j 个人，

- S 到 u_i 连下界为 l_i，上界为 r_i 的边；
- u_i 到 v_j 连容量为 1 的边；
- v_j 到 T 连上下界都为 s_j 的边；
- T 到 S 连容量为 ∞ 的边。

考虑若已知每个人的得分 $s_1 \geq \cdots \geq s_m$ 如何判断可行性，这是上下界可行流的模型：u_i 表示第 i 道题，v_j 表示第 j 个人，

- S 到 u_i 连下界为 l_i，上界为 r_i 的边；
- u_i 到 v_j 连容量为 1 的边；
- v_j 到 T 连上下界都为 s_j 的边；
- T 到 S 连容量为 ∞ 的边。

转成通常的最大流再化简得到：建超级源点 $\mathcal{S}$ 和 $\mathcal{T}$，

- $\mathcal{S}$ 到 S 连容量为 $\sum s_j - \sum l_i$ 的边；
- $\mathcal{S}$ 到 u_i 连容量为 l_i 的边；
- S 到 u_i 连容量为 $r_i - l_i$ 的边；
- u_i 到 v_j 连容量为 1 的边；
- v_j 到 $\mathcal{T}$ 连容量为 s_j 的边。

条件即为最小割 $=\sum s_j$，或者说所有割的方案 $\geq \sum s_j$。设 $\mathcal{S}_u$ 表示被划分到 S 的 u_i 的集合，$\mathcal{T}_u,\mathcal{S}_v,\mathcal{T}_v$ 同理。

- 若 S 划分到 $\mathcal{S}$，则 $|\mathcal{S}_u|\cdot|\mathcal{T}_v|+\sum_{u\in\mathcal{T}_u} r_u \geq \sum_{v\in\mathcal{T}_v} s_v$，所以 $\mathcal{S}_u$ 取 r 比较大的，$\mathcal{T}_v$ 取 s 比较大的;
- 若 S 划分到 $\mathcal{T}$，则 $|\mathcal{S}_u|\cdot|\mathcal{T}_v|+\sum_{v\in\mathcal{S}_v} s_v \geq \sum_{u\in\mathcal{S}_u} l_u$，所以 $\mathcal{S}_u$ 取 l 比较大的，$\mathcal{T}_v$ 取 s 比较大的。

条件即为最小割 $=\sum s_j$，或者说所有割的方案 $\geq \sum s_j$。设 $\mathcal{S}_u$ 表示被划分到 $\mathcal{S}$ 的 u_i 的集合，$\mathcal{T}_u, \mathcal{S}_v, \mathcal{T}_v$ 同理。

- 若 S 划分到 $\mathcal{S}$，则 $|\mathcal{S}_u| \cdot |\mathcal{T}_v| + \sum_{u \in \mathcal{T}_u} r_u \geq \sum_{v \in \mathcal{T}_v} s_v$，所以 $\mathcal{S}_u$ 取 r 比较大的，$\mathcal{T}_v$ 取 s 比较大的；
- 若 S 划分到 $\mathcal{T}$，则 $|\mathcal{S}_u| \cdot |\mathcal{T}_v| + \sum_{v \in \mathcal{S}_v} s_v \geq \sum_{u \in \mathcal{S}_u} l_u$，所以 $\mathcal{S}_u$ 取 l 比较大的，$\mathcal{T}_v$ 取 s 比较大的。

设 $\rho = |\mathcal{S}_u|$，$\sigma = |\mathcal{T}_v|$，将 l, r 也都降序排序，条件即为

$$\rho \cdot \sigma + \sum_{j=\sigma+1}^{m} s_j \geq \max\left\{ t - \sum_{i=\rho+1}^{n} r_i, \sum_{i=1}^{\rho} l_i \right\}$$

可以用斜率优化 $O(n+m)$ 检验。

对于求答案，可以直接二分，考虑判断前 w 名分数相同且 $\geq s$ 是否可行。先排除给定条件中前 w 名有两人分数不同的情况，然后：

- 若已知前 w 名的分数，则先让每人得到尽可能低的分数然后加到总成绩为 t，而后缀和越大越好，所以贪心加到靠后的位置；
- 否则按同样方法做，但若最后填得不满足条件（前 w 名中有分数为 Q 和 $Q-1$ 的）则固定前 w 名分数为 Q 然后再做一遍。

时间复杂度 $O((n+m)(\log n+\log m))$。

给定有限 Abel 群 $G=\prod_{i=1}^{t}\mathbb{Z}_{c_i}$ 的 m 个元素 $g_1,\cdots,g_m$，q 次询问 $[L,R]$ 求 $g_L,\cdots,g_R$ 的生成子群的大小。[4]

$n=\prod_{i=1}^{t}c_i\le 3000$，$m,q\le 10^6$

[4] https://uoj.ac/problem/427

根据中国剩余定理，可以只考虑 c_i 是质数次幂的情形。

类比线性基，考虑维护子群的直积分解式：设当前在第 i 维插入元素 a，

- 若 a 第 i 维的阶不超过线性基第 i 个元素第 i 维的阶，则用这个元素消掉 a 的第 i 维然后插入到更高维；
- 否则把 a 插入线性基，把线性基对应元素 b 弹回，用 a 消掉 b 的第 i 维，然后找到最小的正整数 t 使得 a^t 在第 i 维是 0 并把 a^t 和 b 插入到更高维。

根据中国剩余定理，可以只考虑 c_i 是质数次幂的情形。

类比线性基，考虑维护子群的直积分解式：设当前在第 i 维插入元素 a，

- 若 a 第 i 维的阶不超过线性基第 i 个元素第 i 维的阶，则用这个元素消掉 a 的第 i 维然后插入到更高维；
- 否则把 a 插入线性基，把线性基对应元素 b 弹回，用 a 消掉 b 的第 i 维，然后找到最小的正整数 t 使得 a^t 在第 i 维是 0 并把 a^t 和 b 插入到更高维。

对于求答案，把询问按左端点降序排序，对当前 L 用链表维护可以贡献答案的元素，这样的元素只有 $O(\log n)$ 个。

对每一维的 (x, y) 预处理最小的正整数 t 使得 $y \equiv tx \pmod{c_i}$ 就可以做到插入 $O(\log n)$。

时间复杂度 $O(\sum c_i^2 + m\log^2 n + \mathrm{Sort}(q))$。

给定 n 个不包含重复字符的字符串 t_i 和长为 m 的字符串 s，有 n 个初始为空的字符集合 A_i，每次操作等概率均匀随机选择这 n 个字符串中的一个字符 $t_{i,j}$，将 $t_{i,j}$ 加入 A_i。

求存在 $l \in [0, n-m]$ 使得对 $i \in [1, m]$ 都有 $s_i \in A_{l+i}$ 的期望操作次数 $\bmod 998244353$，需判断无解。T 组数据。[5]

$T \leq 5$，$m \leq n \leq 30$，字符集为小写字母

[5] https://uoj.ac/problem/214

考虑 min-max 容斥，枚举可能出现的子串位置的非空子集 S，贡献为 $(-1)^{|S|-1}\mathbb{E}[\max(S)]$，即为 S 要求的这些字符必须出现，而掷 n 面骰子掷出给定 k 个数字的期望次数为 nH_k。时间复杂度 $O(n2^{n-m})$。

考虑 min-max 容斥，枚举可能出现的子串位置的非空子集 S，贡献为 $(-1)^{|S|-1}\mathbb{E}[\max(S)]$，即为 S 要求的这些字符必须出现，而掷 n 面骰子掷出给定 k 个数字的期望次数为 nH_k。时间复杂度 $O(n2^{n-m})$。

然后考虑优化，当 S 要求出现的字符数相同时贡献也相同，于是可以 dp，设 $f_{i,j,A}$ 表示考虑前 i 个位置，要求出现的字符数为 j，当前第 i 个位置要求出现 $\forall p \in A$，s_p 这些字符，对答案的贡献之和，转移考虑当前位置是否加入 S 即可，初值为 $f_{0,0,\varnothing} = -1$。时间复杂度 $O(n^3 2^m)$。

设字符集 $\Sigma = \{1, 2, \cdots, m\}$，求在所有字符 i 出现不超过 c_i 次的长度为 n 的字符串中，后缀数组的数量 $\bmod 998244353$。[6]

$n, m \le 500$，$0 \le c_i \le n \le \sum c_i$

[6] `https://uoj.ac/problem/199`

引理 (充要条件)

字符串 s 的后缀数组是 $p \iff s_{p_i} \le s_{p_i+1}$ 且当 $p_{i+1}+1$ 在 p_i+1 前面时不取等号。

所以考虑对每种不等式链求对应的后缀数组的数量。

引理 (充要条件)

字符串 s 的后缀数组是 $p \iff s_{p_i} \le s_{p_i+1}$ 且当 $p_{i+1}+1$ 在 p_i+1 前面时不取等号。

所以考虑对每种不等式链求对应的后缀数组的数量。

引理 (bijective proof)

$s \mapsto p$ 是「字符 i 恰出现 c_i 次的字符串」到「对应的不等式链的 $<$ 仅出现在第 $c_1+\cdots+c_i$ 个元素之后的后缀数组」的双射。

所以直接对 $<$ 的位置做子集容斥：设以 $<$ 为分界的每一段长度为 c_i，则枚举断点集合 $\{0=z_0,z_1,z_2,\cdots,z_k=m\}$，贡献系数为

$$\frac{(-1)^{m-k}n!}{\prod(c_{z_{i-1}+1}+\cdots+c_{z_i})!}$$

考虑判断一个不等式链是否满足条件，直接按顺序贪心填尽可能小的字符即可。

设 $f_{i,j,k}$ 表示填了前 i 种字符，已经填了 j 个位置，容斥的上一个断点在 k 的情况下的贡献之和，则初值 $f_{0,0,0}=1$，转移为：

- 第 i 种字符都在一段，且不是这段的最后一种字符：$f_{i,j,k} \leftarrow f_{i-1,j-c_i,k}$；
- 第 i 种字符是这段的最后一种字符，且后面的 < 是断点：$f_{i,j,j} \leftarrow f_{i-1,j-l,k}/(j-k)!\ (1 \le l \le c_i)$；
- 第 i 种字符是这段的最后一种字符，且后面的 < 不是断点：$f_{i,j,k} \leftarrow -f_{i-1,j-l,k}\ (1 \le l \le c_i)$；

答案即为 $n!\sum_{i=1}^{m} f_{i,n,n}$，使用前缀和优化，时间复杂度 $O(n^2m)$。

维护长为 n 的字符串，字符集 $\Sigma = [-10^9, 10^9] \cap \mathbb{Z}$，$q$ 次操作:

- 给定 l, r, d，$\forall i \in [l, r]$，令 $s_i := s_i + d$;
- 给定 l, r，求 $s[l..r]$ 的最小后缀。[7]

$n \le 2 \cdot 10^5$，$q \le 3 \cdot 10^4$

[7] https://uoj.ac/problem/296

定义 (*Significant Suffixes*)

对于字符串 $s = uv$，若存在字符串 t 使得 vt 是 st 的最小后缀，则称 v 是 s 的**关键后缀**。

定理 (*Significant Suffixes Log Theorem*)

对于长为 n 的字符串 s，关键后缀的数量不超过 $\log n$。

证明.

设 u, v 是两个关键后缀，不妨设 $|u| < |v|$，则 u 是 v 的 border，若 $|v| \le 2|u|$，则 v 有长度为 $|v| - |u| \le |v|/2$ 的周期 α，即 $u = \alpha\beta$，$v = \alpha^2\beta$。设字符串 t 使得 ut 是 st 的最小后缀，则 $ut < vt$，所以 $\beta t < \alpha\beta t = ut$，与 u 是关键后缀相矛盾，所以 $|v| > 2|u|$。 $\square$

线段树维护区间关键后缀，考虑如何合并：左儿子只能留一个，即在只考虑到右端点的情况下最小的那个，若范围内比较不出来就选下标最小的。

使用分块维护 hash 值，时间复杂度 $O(n\log^2 n + q(\log^3 n + \sqrt{n}))$。

定义 (*Deterministic Finite Automaton*)

一个**确定性有限状态自动机** (DFA) 意指以下资料：

- 有限集合 Q，其元素称为**状态**；
- 有限集合 Σ，其元素称为**字符**；
- $\delta : Q \times \Sigma \to Q$，称为**转移函数**；
- $q_0 \in Q$，称为**初始状态**；
- $F \subset Q$，称为**接受状态集合**。

定义 Kleene 星号算子如下：

$$V^* := \bigcup_{n=0}^{+\infty} V^n = \{\epsilon\} \cup V \cup V^2 \cup V^3 \cup \cdots$$

其中 ϵ 表示空序列。Σ^* 即为所有 Σ 上字符串的集合。

定义 (*Regular Language*)

递归定义**扩展转移函数** $\delta^* : Q \times \Sigma^* \to Q$ 如下:

- $\delta^*(q, \epsilon) = q$，$\forall q \in Q$;
- $\delta^*(q, u\sigma) = \delta(\delta^*(q, u), \sigma)$，$\forall q \in Q, u \in \Sigma^*, \sigma \in \Sigma$

设 $M = (Q, \Sigma, \delta, q_0, F)$ 是一个 DFA，若 $w \in \Sigma^*$ 满足 $\delta^*(q_0, w) \in F$ 则称 M **接受** w。

称 Σ^* 的子集为**形式语言**，令语言 $L(M) := \{w \mid M \text{ 接受 } w\}$，则称 M **识别** $L(M)$，能被 DFA 识别的语言称为**正则语言**，若两个 DFA 识别相同的语言则称它们等价。

[8] https://uoj.ac/problem/375

定义 (*Regular Language*)

递归定义**扩展转移函数** $\delta^* : Q \times \Sigma^* \to Q$ 如下:

- $\delta^*(q, \epsilon) = q$，$\forall q \in Q$;
- $\delta^*(q, u\sigma) = \delta(\delta^*(q, u), \sigma)$，$\forall q \in Q, u \in \Sigma^*, \sigma \in \Sigma$

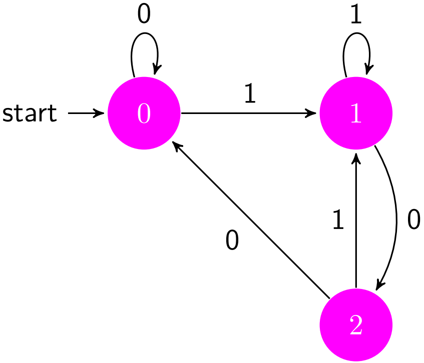
设 $M = (Q, \Sigma, \delta, q_0, F)$ 是一个 DFA，若 $w \in \Sigma^*$ 满足 $\delta^*(q_0, w) \in F$ 则称 M **接受** w。

称 Σ^* 的子集为**形式语言**，令语言 $L(M) := \{w \mid M \text{ 接受 } w\}$，则称 M **识别** $L(M)$，能被 DFA 识别的语言称为**正则语言**，若两个 DFA 识别相同的语言则称它们等价。

给定正整数 m, k，字符集 $\Sigma = \{0, 1, \cdots, m-1\}$，定义语言 $\mathcal{L}$ 表示所有 m 进制意义下是 k 的倍数的字符串。求能识别 $\mathcal{L}$ 的 DFA 的状态数最小值。T 组数据。[8] $T \leq 3 \cdot 10^5$，$m, k \leq 10^{18}$

[8] `https://uoj.ac/problem/375`

显然答案存在且不会超过 k：取状态为 m 进制意义下模 k 的余数。但是可能有更优的答案，比如 $m=2$，$k=4$ 时就只需要 3 个点~~那咋办呢 qwq~~



定义 (*Minimization of DFA*)

对于 DFA 的两个状态 p,q，定义等价关系 $p \sim q$ 当且仅当对于所有 $w \in \Sigma^*$ 都有 $\delta^*(p,w) \in F \iff \delta^*(q,w) \in F$。

剔除初始状态无法到达的状态后，设状态的等价类为 $\{S_1, \cdots, S_m\}$，考虑构造一个新的 DFA M'：

- 将 M 中的等价类 S_i 作为 M' 中的状态 i；
- 对于等价类 S_i，若存在 $u \in S_i$ 和 $c \in \Sigma$ 使得 $\delta_M(u,c) \in S_j$，则对于任意 $v \in S_i$ 都有 $\delta_M(v,c) \in S_j$，所以可定义 $\delta_{M'}(i,c) := j$；
- 对于等价类 S_i 的所有状态，要么均为接受状态，要么均不是接受状态，所以可同理定义 M' 的接受状态；
- 令包含初始状态的等价类 S_i 对应的状态 i 为 M' 的初始状态。

容易验证新 DFA M' 与原 DFA M 等价，M' 即称为 M 的**最小化**。

对于语言 $\mathcal{L}$ 和 $x, y \in \Sigma^*$，定义等价关系 $x \equiv_{\mathcal{L}} y$ 当且仅当对于所有 $z \in \Sigma^*$ 都有 $xz \in \mathcal{L} \iff yz \in \mathcal{L}$。

定理 (*Myhill-Nerode*)

语言 $\mathcal{L}$ 是正则的 $\iff \equiv_{\mathcal{L}}$ 的等价类个数有限，此时描述 $\mathcal{L}$ 的最小 *DFA* 在同构意义下唯一，且它的状态与 $\equiv_{\mathcal{L}}$ 的等价类相对应。

对于语言 $\mathcal{L}$ 和 $x, y \in \Sigma^*$，定义等价关系 $x \equiv_{\mathcal{L}} y$ 当且仅当对于所有 $z \in \Sigma^*$ 都有 $xz \in \mathcal{L} \iff yz \in \mathcal{L}$。

定理 (*Myhill-Nerode*)

语言 $\mathcal{L}$ 是正则的 $\iff \equiv_{\mathcal{L}}$ 的等价类个数有限，此时描述 $\mathcal{L}$ 的最小 *DFA* 在同构意义下唯一，且它的状态与 $\equiv_{\mathcal{L}}$ 的等价类相对应。

证明.

显然对任意的 DFA A，$L(A)$ 的等价类个数不大于 A 的状态数。
而对等价类个数有限的语言 $\mathcal{L}$，可以构造 DFA A 识别 $\mathcal{L}$：初始状态对应空串的等价类，接受状态对应全体元素属于 $\mathcal{L}$ 的等价类，转移边直接合并。
若存在与 A 不同构的 DFA B 识别 $\mathcal{L}$ 且状态数不超过等价类个数，则存在不等价的串 x, y 到达 B 的同一个状态，矛盾。 $\square$

所以题目所求即为给定 DFA 求最小化。

例 (等价类划分算法)

定义 $p \sim_k q$ 表示对于任意长度 $\leq k$ 的字符串 w 都有 $\delta^*(p, w) \in F \iff \delta^*(q, w) \in F$，设等价类集合为 Π_i，则 $\Pi_0 = \{F, Q \setminus F\}$，考虑由 Π_i 推出 Π_{i+1}：

枚举 $c \in \Sigma$ 和 $S \in \Pi_i$，若存在 $u, v \in S$ 使得 $\delta(u, c)$ 和 $\delta(v, c)$ 属于 Π_i 的不同组，则按照转移后的组把 S 划分。若不能再划分则 Π_i 即为所求的等价类集合。

时间复杂度 $O(n^2|\Sigma|)$，在随机数据下会表现得更好。

所以题目所求即为给定 DFA 求最小化。

例 (等价类划分算法)

定义 $p \sim_k q$ 表示对于任意长度 $\le k$ 的字符串 w 都有 $\delta^*(p, w) \in F \iff \delta^*(q, w) \in F$，设等价类集合为 Π_i，则 $\Pi_0 = \{F, Q \setminus F\}$，考虑由 Π_i 推出 Π_{i+1}：

枚举 $c \in \Sigma$ 和 $S \in \Pi_i$，若存在 $u, v \in S$ 使得 $\delta(u, c)$ 和 $\delta(v, c)$ 属于 Π_i 的不同组，则按照转移后的组把 S 划分。若不能再划分则 Π_i 即为所求的等价类集合。

时间复杂度 $O(n^2|\Sigma|)$，在随机数据下会表现得更好。

~~可以获得 30 分的好成绩，但是这和正解有什么关系呢 w~~

对于两个状态 x, y，若 $x \cdot m^i \equiv y \cdot m^i \pmod{k}$ 且 x 和 y 在 i 步之内都到不了 0，那么 x 就与 y 等价。

对 i 递推，设 $x \cdot m^i$ 的上界为 l，模数为 k，$d = \gcd(m, k)$，若 $d = 1$ 或 $l \le k/d$ 则无法再合并，否则上界变为 $k - m(k - l)$（若 ≤ 0 也直接结束），合并之后都除以 d，多出来的 $m(k - l)/d$ 个组不会再合并，直接贡献答案。

这是一道通信题

使用至多 12500 个 bit 编码 1024 个元素的 32 位整数到 10 位整数的 map。[9]

思考题：怎么用 32768 个 bit 编码 1024 个元素的 32 位整数到 32 位整数的 map?

[9] https://uoj.ac/problem/178

随机 hash 函数把 32 位压缩到 17 位，然后对每个值记录哪些键与其对应，可以做到 19×1024 个 bit，期望得分 65。

随机 hash 函数把 32 位压缩到 17 位，然后对每个值记录哪些键与其对应，可以做到 19×1024 个 bit，期望得分 65。

随机 hash 函数把键值均匀分成两组，把随机次数传过去然后分治，递归到只有 8 个键值的时候用随机 hash 函数将其压缩到排列。

传不定长整数的小技巧：取初始长度 l，若位数 $\leq l$ 则输出 1 然后输出 l 位，否则先输出 k 个 0 然后输出 $l+k$ 位。

这是一道交互题

交互库有 n 个点的完全图，每条边染红色或蓝色，已知 $\forall i \in [1,k)$，i 到 $i+1$ 的连边为红色，你可以询问 $2k$ 次某条边的颜色，求一条 $k+1$ 个点的简单路径，使得所有边的颜色相同。[10]

$k \le 2000$，$\frac{3k}{2} \le n \le 2k$

[10] https://uoj.ac/problem/486

设 $2, \cdots, k-1$ 为红点，$k+1, \cdots, n$ 为蓝点。

称一条路径 C 为**交错路**当且仅当所有边都是蓝色，且两端点都是蓝点，且红蓝点交错。其大小 $|C|$ 为路径上的点数。

设 $2, \cdots, k-1$ 为红点，$k+1, \cdots, n$ 为蓝点。

称一条路径 C 为**交错路**当且仅当所有边都是蓝色，且两端点都是蓝点，且红蓝点交错。其大小 $|C|$ 为路径上的点数。

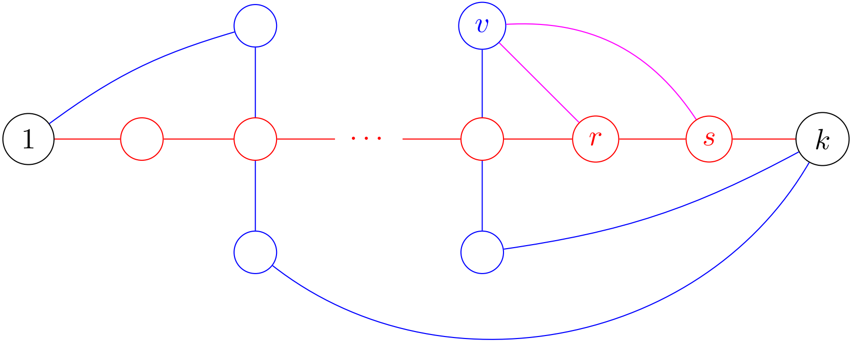
考虑两条不相交的交错路 A, B，若 $|A|+|B|>k-2$，某个端点到 1 或 k 的连边为红色就直接做完了，否则可以找到至少 $k+1$ 个点的蓝圈，也做完了。

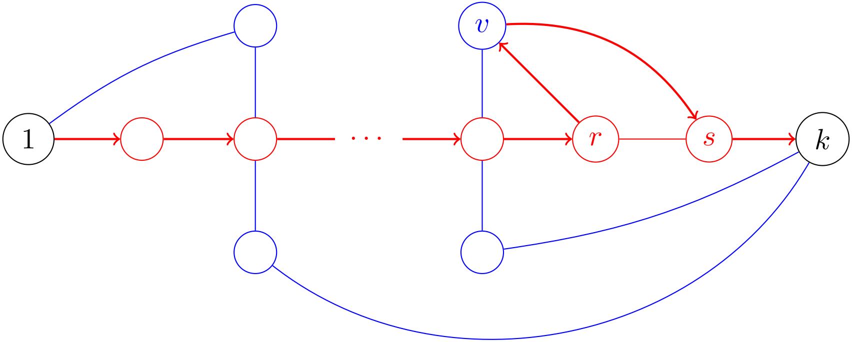
设 $2, \cdots, k-1$ 为红点，$k+1, \cdots, n$ 为蓝点。

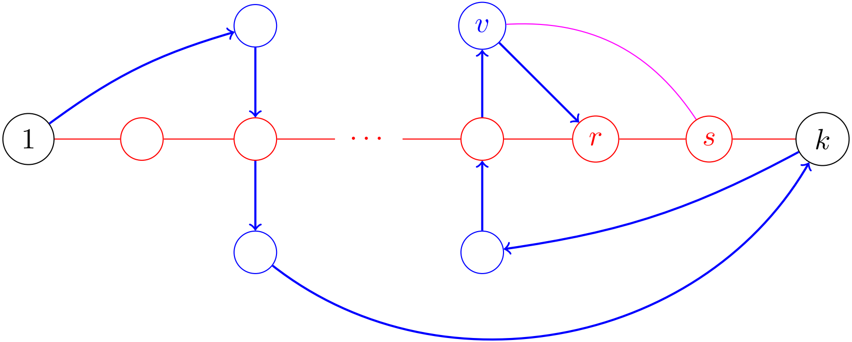
称一条路径 C 为**交错路**当且仅当所有边都是蓝色，且两端点都是蓝点，且红蓝点交错。其大小 $|C|$ 为路径上的点数。

考虑两条不相交的交错路 A, B，若 $|A|+|B|>k-2$，某个端点到 1 或 k 的连边为红色就直接做完了，否则可以找到至少 $k+1$ 个点的蓝圈，也做完了。

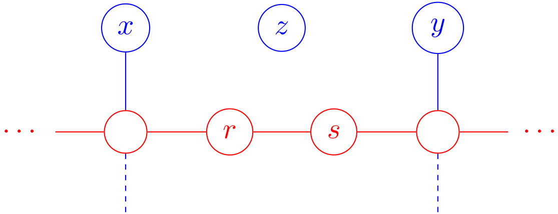
若 $|A|+|B|=k-2$，由交错路的定义知 $k-2$ 个红点被占用了 $\frac{k-2}{2}-1$ 个，由抽屉原理知存在没被占用的相邻红点 r, s。设蓝圈上的一个蓝点为 v，讨论一下 (r, v) 和 (s, v) 的颜色即可。


1
⋯
v
r
s
k


v
1
· · ·
r
s
k

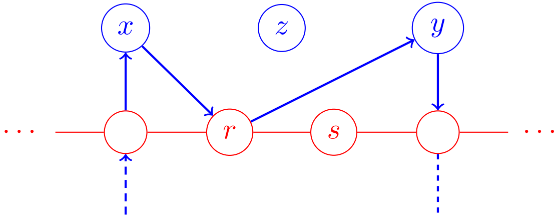

1
v
r
s
k
· · ·

若 $|A|+|B|<k-2$，设 A,B 的其中一个端点分别为 x,y，则 $|A\cup B\cup\{1,2,\cdots,k-1,k\}|<\frac{3}{2}k$，所以存在 A,B 之外的蓝点 z，同理有未占用的相邻红点 r,s。



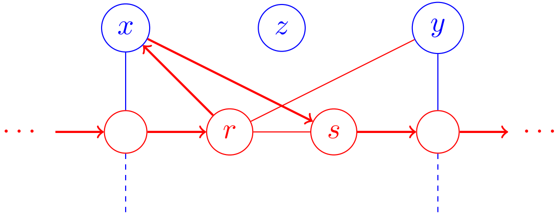
若 $|A|+|B|<k-2$，设 A,B 的其中一个端点分别为 x,y，则 $|A\cup B\cup\{1,2,\cdots,k-1,k\}|<\frac{3}{2}k$，所以存在 A,B 之外的蓝点 z，同理有未占用的相邻红点 r,s。

若 $(x,r),(y,r)$ 都为蓝色（单个蓝点也是交错路）：



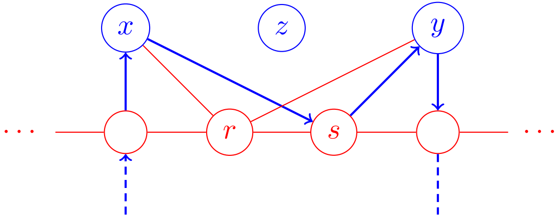
若 $|A|+|B|<k-2$，设 A,B 的其中一个端点分别为 x,y，则 $|A\cup B\cup\{1,2,\cdots,k-1,k\}|<\frac{3}{2}k$，所以存在 A,B 之外的蓝点 z，同理有未占用的相邻红点 r,s。

若 $(x,r),(y,r)$ 都为红色，则查询 (x,s) ：



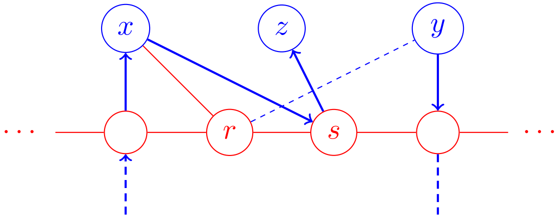
若 $|A|+|B|<k-2$，设 A,B 的其中一个端点分别为 x,y，则 $|A\cup B\cup\{1,2,\cdots,k-1,k\}|<\frac{3}{2}k$，所以存在 A,B 之外的蓝点 z，同理有未占用的相邻红点 r,s。

若 $(x,r),(y,r)$ 都为红色，则查询 $(x,s),(y,s)$：



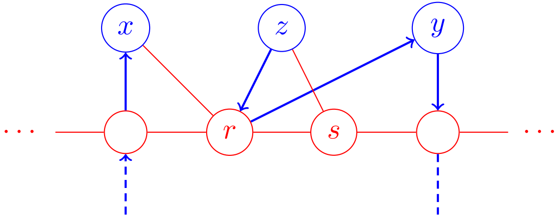
若 $|A|+|B|<k-2$，设 A,B 的其中一个端点分别为 x,y，则 $|A\cup B\cup\{1,2,\cdots,k-1,k\}|<\frac{3}{2}k$，所以存在 A,B 之外的蓝点 z，同理有未占用的相邻红点 r,s。

若 (x,r) 为红色，$(y,r),(z,s)$ 为蓝色，则查询 (x,s)：



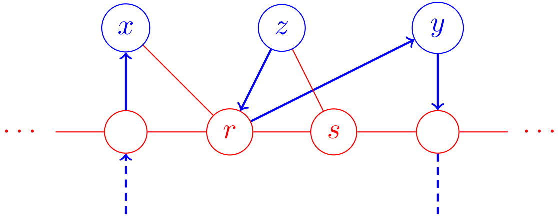
若 $|A|+|B|<k-2$，设 A,B 的其中一个端点分别为 x,y，则 $|A\cup B\cup\{1,2,\cdots,k-1,k\}|<\frac{3}{2}k$，所以存在 A,B 之外的蓝点 z，同理有未占用的相邻红点 r,s。

若 $(x,r),(z,s)$ 为红色，(y,r) 为蓝色，则查询 (z,r)：



若 $|A| + |B| < k - 2$，设 A, B 的其中一个端点分别为 x, y，则 $|A \cup B \cup \{1, 2, \cdots, k-1, k\}| < \frac{3}{2}k$，所以存在 A, B 之外的蓝点 z，同理有未占用的相邻红点 r, s。

若 $(x, r), (z, s)$ 为红色，(y, r) 为蓝色，则查询 (z, r)：



从 A, B 都是单个蓝点开始，4 次询问使得 $|A| + |B|$ 增加 2，最后 5 次询问合并 A 和 B，总询问次数不超过 $2k - 3$。

求有多少个本质不同的 $\{1, 2, \cdots, n\}$ 的子集构成的多重集 L，满足「对所有 $\{1, 2, \cdots, n\}$ 的排列 σ，将 L 的元素作置换得到的 $\sigma(L)$ 两两不同」且 $|L|$ 尽量小。

若存在排列 σ 使得 $L_1 = \sigma(L_2)$ 则认为 L_1 与 L_2 是本质相同的。若答案 > 1000 则输出 -1。[11]

$2 \leq n \leq 2 \cdot 10^{18}$

[11] https://atcoder.jp/contests/agc044/tasks/agc044_f

先考虑 L 的元素不重复的情况。设 $k=|L|$，利用 $k\times n$ 的 01 矩阵 T 描述，定义满足条件的 L 对应的矩阵 T 是**好的**。

设 σ,τ 分别是 $[k]$ 和 $[n]$ 的排列，$\sigma\circ T$ 和 $T\circ\tau$ 分别表示对行/列作置换，则 T 是好的当且仅当 $\sigma\circ T=T\circ\tau \implies \sigma$ 和 τ 都是恒等排列蕴含没有两行/列完全相同

引理 (对称性)

定义 T^t 表示矩阵转置，T^c 表示对列的集合取补集得到的 $k \times (2^k - n)$ 的矩阵，则 T 是好的 $\iff T^t$ 是好的 $\iff T^c$ 是好的。

证明.

若 T 是好的，因为条件对行、列对称所以 T^t 也是好的，而 σ 对 $\{0,1\}^k$ 的置换作用构成排列，所以条件等价于 σ 对列集合的置换作用不映射到自身，则补集也满足条件，所以 T^c 也是好的。 □

设 $g(n)$ 表示最小的 k 使得存在 $k \times n$ 的好矩阵。

则有不等式 $2^{g(n)} - n \geq g(g(n))$，归纳定义 $G(n)$ 如下：

$$G(n) = \begin{cases} 0, & n = 1, \\ \min\{k \in \mathbb{N}_+ \mid 2^k - n \geq G(k)\}, & n > 1. \end{cases}$$

引理

存在 $k \times n$ 的好矩阵 $\iff G(n) \le k \le 2^n - G(n)$
蕴含 $g(n) = G(n)$

证明.

考虑对 n 归纳。若 $k < n$，则 $k \ge G(n) \iff n \le 2^k - G(k) \iff$ 存在 $n \times k$ 的好矩阵。

若 $k \ge n$，由对称性只需证 $k \le 2^{n-1}$ 的情况：
$\{\{1\},\{1,2\},\cdots,\{n-1,n\}\}$ 加上 $k-n$ 个大小 >2 的集合。 □

设 $c(k,n)$ 表示 $k\times n$ 的好矩阵个数，所求即为 $c(g(n),n)$。

引理

若 $6\le k\le n\le 2^{k-1}$，则 $c(k,n)>1000$。

证明.

对于如下满足条件的两个方案:

$$\{\{1\},\{1,2\},\cdots,\{k-3,k-2\},\{k-2,k-1\}^c\}$$
$$\{\{1\}^c,\{1,2\}^c,\cdots,\{k-3,k-2\}^c,\{k-2,k-1\}\}$$

可以任意添加大小不是 2 和 $k-2$ 的集合，所以有

$$c(k,n)\ge 2\binom{2^k-2\binom{k}{2}}{n-k+2}\ge 34\cdot 33>1000.$$

然后考虑可重的情况即存在两行完全相同，则有 $2^{k-1} \geq n$。

由 $G(n)$ 定义知 $k \geq G(n)$，所以行数最小值不变。

若 $k = G(n)$ 则 $2^{G(n)-1} \geq n$，由 $G(n)$ 定义知存在 $m \in \mathbb{N}_+$ 使得 $2^m - G(m) < n \leq 2^m$，结合上述放缩可知 $n = 4, 7, 8$。

~~爆搜就完事了 /kx~~

给定 n 个点的树，每个点有初始为 1 的点权，每次操作选择相邻点权之和为奇数的点 v，把 v 的点权取反（0 变为 1，1 变为 0），求能得到的点权状态数 $\bmod 998244353$。[12]

$n \leq 2 \cdot 10^5$

[12] https://atcoder.jp/contests/agc052/tasks/agc052_f

因为是数状态数，考虑怎样的状态可以被全 1 到达。又因为操作可逆所以考虑怎样的状态可以到达全 1。

发现操作不改变点权为 1 的点的连通分量个数的奇偶性。

因为是数状态数，考虑怎样的状态可以被全 1 到达。又因为操作可逆所以考虑怎样的状态可以到达全 1。

发现操作不改变点权为 1 的点的连通分量个数的奇偶性。

根据 AGC 套路猜想这也是充分条件，但是必不会有这么简单（

定理 (充要条件)

若点权为 1 的点的连通分量个数为大于 1 的奇数，且存在某个点的相邻点权之和为奇数即可以操作，且存在某个点的度数 ≥ 3 且其中至少 2 个子树大小 ≥ 2，则可以使得连通分量个数减少从而到达全 1 的状态。

首先可以将每个点权为 1 的连通分量压缩为一个点的权值为 1 而其他为 0，此时点权为 1 的点构成独立集。

定义合并操作选择点权为 0 且相邻点权之和为奇数的点 v，将 v 改为 1 而把 v 相邻的 1 改为 0。

引理

若可以改变某个点的权值，则所有点也都可以。

证明.

对点数归纳，设 u 的儿子为 $x_1, \cdots, x_k$。

- 若 u 的点权为 0，当 x_i 的点权之和为偶数时，由归纳假设知可以改变某个 x_i 的点权。从而对 u 使用合并操作改变 u 的点权；
- 若 u 的点权为 1，则 $x_1, \cdots, x_k$ 的点权都为 0，当所有 x_i 的儿子的点权之和为奇数时，由归纳假设知可以改变某个 x_i 的儿子的点权。从而对 x_i 使用合并操作改变 u 的点权。 □

引理

若存在点权为 1 但无法操作，则给点权为 0 的点 u 加上点权为 1 的叶子 u_1 之后可以合并 ≥ 3 个 1。

设定理条件所设为点 v，若 v 的点权为 1 则由引理将其反转，设儿子为 $x_1, \cdots, x_k$，若点权之和 ≥ 3 则由引理将其合并到 v，否则若

- 恰有 x_1 的点权为 1，若某个 x_2 的子树内有 1，则将 x_1 移到 v 之后由引理 2 做合并操作，否则可以将 x_1 移到另一个儿子 x_2，又因为 x_1 的子树内还有其他的 1 所以由引理将 x_1 反转，转化为下一种情况；
- 恰有 x_1, x_2 的点权为 1，同理其他的子树没有 1，不妨设可以把 x_1 的点权反转，则可以把 x_2 移到除 x_1 之外任意一个儿子，不妨设 x_2 的子树大小 > 1，则把 x_2 移到更深，把 x_1 移到另一个子树，同理再将 x_1 反转，此时 v 的儿子点权之和为 3。

对于两端扫帚，设链长为 l，两端 A, B 分别连出 x, y 个叶子，

- 若 A, B 点权为 1，则叶子随便选，链上都是 1，共 2^{x+y} 种情况；
- 若 A 点权为 1，B 点权为 0，则 A 的叶子随便选，B 的叶子恰有偶数个 1，链上恰有 A 一侧的前缀为 1，共 $2^{x+y-1}l$ 种情况；
- 若 A, B 点权都为 0，所有叶子恰有奇数个 1，则链上全是 0，共 2^{x+y-1} 种情况；
- 若 A, B 点权都为 0，A, B 的叶子分别有偶数个 1，则链上恰有一个连续段为 1，共 $2^{x+y-2}\binom{l}{2}$ 种情况。

其他情况可以直接 dp，时间复杂度 $O(n)$。