第十一届中国大学生程序设计竞赛（济南）题解

中山大学命题组

2025-11-16

Problem C. 查找关键词

题意简述

给定一个长度为 m 的排列 b，另外给出 n 个序列，第 i 个序列的长度为 l_i，且仅包含 $[1, 10^6]$ 范围内的正整数。

对每个序列，需要选出尽可能多**位置不交**的子序列，使得每个子序列都恰好等于排列 b。输出最多的可选子序列的个数。

数据范围：$m \leq 10, \sum l_i \leq 10^6$。

Problem C. 查找关键词

本题的基础思路是贪心，具体包含两种方案。

方案 1：贪心选取子序列，直到无法选取为止。具体的，考虑选取序列中第一个未被选择的 b_1，然后选择在其后面第一个未被选择的 b_2，以此类推。如果最终能选满 m 个数，则选择了一个子序列，否则结束流程。使用 `set` 等 STL 即可实现 $\mathcal{O}(\sum l_i \log l_i)$ 的时间复杂度，进一步利用单调性可以做到 $\mathcal{O}(\sum l_i)$，此处不再展开。

Problem C. 查找关键词

方案 2：按顺序枚举序列的所有值，当枚举到一个值 x 的时候，如果 $x > m$，忽略即可。否则，假设 x 出现在排列 b 的第 p 个位置，那么 x 唯一的贡献是将一个已经匹配了前 $p-1$ 个元素的子序列之后追加一个 x。因此，可以维护 f_i 代表当前匹配了前 i 个元素的子序列个数，初始令 $f_0 = +\infty, f_{[1,m]} = 0$，每次尝试从 f_{p-1} 减去 1 并转移到 f_p 即可。时间复杂度为 $\mathcal{O}(\sum l_i)$。

Problem K. 开火车

题意简述

Alice 和 Bob 正在游玩开火车，牌组是 $1 \sim n$ 的牌各两张，每人 n 张牌。二人轮流出牌，Alice 先手，每个人会选择一张牌放在牌堆的顶部，如果牌堆出现了两张牌，那么这两张牌和中间的部分会被收走，并且当前玩家得一分。

给出 Alice 的出牌顺序，构造 Bob 的出牌顺序，从而最小化 Alice 的分数。

数据范围：$n \leq 5 \times 10^5$。

Problem K. 开火车

可以发现，如果 Alice 出的第一张牌是 Bob 没有的牌，那么 Bob 无论如何也无法阻拦 Alice 使用这一张牌获得 1 分。可以通过后续的构造方案证明，除了这一得分可能性之外，Bob 总是能阻止 Alice 得分，因此“分数最小值”的答案为：如果 Alice 出的第一张牌是 Bob 没有的牌，则为 1，否则为 0。

Problem K. 开火车

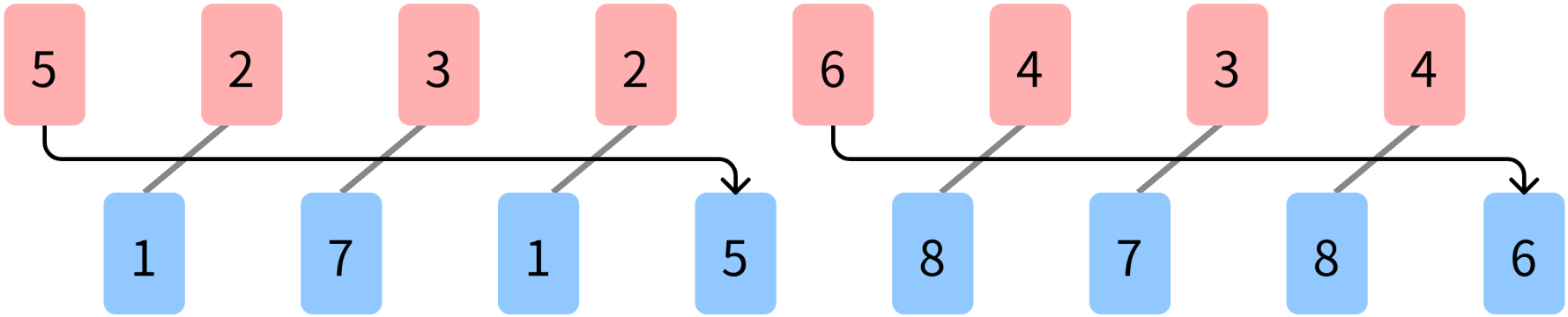
需要注意到：在 Alice 拥有两张一样的牌时，Bob 也可以选择两张一样的牌与之对应。对于 Alice 的两张 1，如果 Bob 使用两张 2 对应，那么一种可行的方案就是将两张 2 分别在两张 1 被打出之前的一轮打出。此时在 Bob 打出第二张 2 时：

- 如果第一张 2 被收走，那么可以证明 Alice 的第一张 1 也被收走，此时 Alice 的第二张 1 无法得分；
- 否则，第一张 2 没有被收走，那么 Bob 得一分，并且两张 2 之间的所有牌也都会被收走。这些牌中包含了第一张 1，那么 Alice 的第二张 1 也是无法得分的。

因此，对 Alice 打出的两张一样的牌，只要没有在第一轮打出，即可使用 Bob 已有的两张一样的牌阻止 Alice 得分。最后按照顺序处理 Bob 剩余的牌即可。

时间复杂度为 $\mathcal{O}(n)$。

Problem K. 开火车



Problem L. 列队

题意简述

给定一个长度为 n 的排列 a_i，需要支持 m 次操作，每次操作是如下两种操作之一：

- 给定 $1 \le x < y \le n$，交换 a_x 和 a_y；
- 给定 $1 \le l \le r < n$，按照如下方式对序列 a_i 模拟 $n-1$ 次操作：
  - 从序列 a_i 的左侧拿出两个值，将较大的一个取走，较小的一个放到序列的右侧。

  求第 l 次到 r 次之间取走的数字总和。

数据范围：$n, m \le 10^5$。

Problem L. 列队

假设第 i 次操作中被放到序列右侧的值等于 a_{n+i}。可以得到：第 i 次操作中有 $a_{n+i} = \min(a_{2i-1}, a_{2i})$，取出的数字为 $\max(a_{2i-1}, a_{2i}) = a_{2i-1} + a_{2i} - a_{n+i}$。

考虑一个包含 $1 \sim 2n-1$ 数字的图，对所有 $1 \le i < n$ 连边 $n+i \to 2i-1$ 与 $n+i \to 2i$，代表依赖关系。假设 $i = n - j$，带入后发现连边变为 $2n - j \to 2n - (2j+1)$ 和 $2n - j \to 2n - 2j$，恰好对应线段树的连接方式。据此可以发现这个图实际上形成了类似线段树的结构，那么 a_i 的修改只会影响 $\mathcal{O}(\log n)$ 个位置。

因此对修改可以通过上述连边方式暴力跳边，将取出数字的新值在支持单点修改和区间和查询的数据结构上进行对应操作即可。共有 $\mathcal{O}(\log n)$ 个需要修改的位置，使用线段树或者树状数组即可在 $\mathcal{O}(m \log^2 n)$ 的时间复杂度内解决这个问题。

Problem F. 方格填数

题意简述

定义一个序列的连续度为其极长同值连续段长度的平方和。给定 n 个整数上的区间 $[l_i, r_i]$，求所有满足 $x_i \in [l_i, r_i]$ 的序列 x 的连续度最小值。

数据范围：$n \le 10^6$，$1 \le l_i \le r_i \le 10^9$。

Problem F. 方格填数

本题的实现较为多样，存在仅使用动态规划的做法，主要利用的性质是：考虑下标在 $[1, i]$ 范围内的区间对应的最小连续度方案，第 i 个区间中只有至多三个值是有用的。下面主要介绍基于贪心的做法。

长度至少为 3 的区间总能找到不等于左右两个数字的值，因此可以让长度为 3 的区间将区间序列分为若干段，其中每一段只包含长度为 1 或者 2 的区间。

对每一段内部，如果存在相邻两个区间，满足二者的交集为空，那么同样可以从这两个区间的中间断开，形成更小的子问题。现在，所有的子问题都满足区间长度至多为 2 并且相邻两个区间有交。接下来在上述限制下考虑原问题。

Problem F. 方格填数

首先可以发现，连续的长度为 1 的区间显然都是是相同的，而连续的长度为 2 的区间需要进行决策。不妨假设这些长度为 2 的区间对应的下标范围是 $[p, q]$。

一种可能的情况是：存在一种决策，对任意 $i \in [p-1, q]$ 都有 $x_i \neq x_{i+1}$。这个部分可以通过线性检查确认。

进一步考虑无法满足上述条件的情况，此时如果 $p \neq q$，也就是中间存在至少两个区间，可以证明最优策略是在中间存在恰好一个 $i \in [p, q)$，使得 $x_i = x_{i+1}$，也就是中间出现了一个长度为 2 的极长同值连续段。最终剩余的情况就是 $p = q$。

最后一个没有被讨论过的情况就是：两个长度为 1 的区间中间夹着一个长度为 2 的区间，并且中间区间的决策要么和左侧值相同，要么和右侧值相同。样例的第三组测试数据对应的就是这样的情况。

Problem F. 方格填数

多个类似的情况之间会互相影响，可以使用动态规划处理，不过更轻松的方法依然是使用贪心：

- 从左往右考虑每个长度为 2 的区间，如果其左边的连续段长度不大于右边的连续段长度，则选择左边的连续段，否则选择右边的连续段。

这一贪心的正确性证明在此略去。最终的时间复杂度为 $\mathcal{O}(n)$，并且常数与动态规划方法相比较小。

Problem A. 暗语

题意简述

共 T 次询问，一次询问给定两个在 $[0, 2^{64})$ 范围内的整数 a, b，求最小的非负整数 x 使得 $a^x \equiv b \pmod{2^{64}}$，或者指出整数 x 不存在。

数据范围：$T \leq 10^5$。

Problem A. 暗语

首先考虑 a 为偶数的情况。需要注意到 $a^{64} \equiv 0 \pmod{2^{64}}$，那么只需要检测 $x \in [0, 64]$ 即可。接下来考虑 a 是奇数的情况。

对于奇数 a，欧拉定理告诉我们 $a^{\varphi(2^{64})} = a^{2^{63}} \equiv 1 \pmod{2^{64}}$。可以证明，满足 $a^k \equiv 1 \pmod{2^{64}}$ 的最小 k 一定是 2 的幂次。k 也被称为 a 的阶。

在阶的性质下可以发现，$a^0, a^1, a^2, ..., a^{k-1}$ 对 2^{64} 取模的结果互不相同（否则可以选择两个相同的元素相除得到更小的阶，矛盾）。因此，若答案存在，则必然只存在恰好一个 $x \in [0, k)$，使得 $a^x \equiv b \pmod{2^{64}}$。

考虑构造。实际上，欧拉定理告诉我们对任意 $k \geq 1$，总有 $a^{\varphi(2^k)} = a^{2^{k-1}} \equiv 1 \pmod{2^k}$。这启发我们按位构造。

Problem A. 暗语

定义 $\mathrm{ctz}(x)$ 表示 x 的二进制低位连续 0 的个数。考虑一个奇数 u，假设 $k = \mathrm{ctz}(u-1)$，另外有 $u = 1 + p2^k$，其中 p 是奇数。注意到

$$u^2 = \left(1 + p2^k\right)^2 = 1 + 2p2^k + p^2 2^{2k} = 1 + p2^{k+1} + p^2 2^{2k}$$

可以发现 $\mathrm{ctz}(u^2 - 1) \geq k + 1 > \mathrm{ctz}(u-1)$。同理考虑 $a^{2^0} - 1, a^{2^1} - 1, a^{2^2} - 1, \ldots$，它们的 ctz 必然是严格单调递增的。

基于这一性质可以给出如下构造：从 0 开始枚举 k，计算 a^{2^k} 的值。如果为 1，则检验构造的结果 x 是否满足条件并退出。否则，考虑当前 a^x 和 b 的第 $\mathrm{ctz}\left(a^{2^k} - 1\right)$ 位，如果不相同，则需要让前者乘以 a^{2^k} 进行调整，也就是令 $x \leftarrow x + 2^k$。

通过上述方式即可保证构造正确。时间复杂度为 $\mathcal{O}(n \log V)$ 或 $\mathcal{O}(n \log^2 V)$，其中 $V = 2^{64}$ 代表值域。

Problem E. 二分图问题

题意简述

给定一棵包含 n 个节点的无根树，树上每个点有一个颜色 c_i。对两个点集 S 和 T，称有序对 (S,T) 是好的，当且仅当：

- S 和 T 各自包含的点颜色相同；
- 包含 S 的最小连通子图和包含 T 的最小连通子图不交。

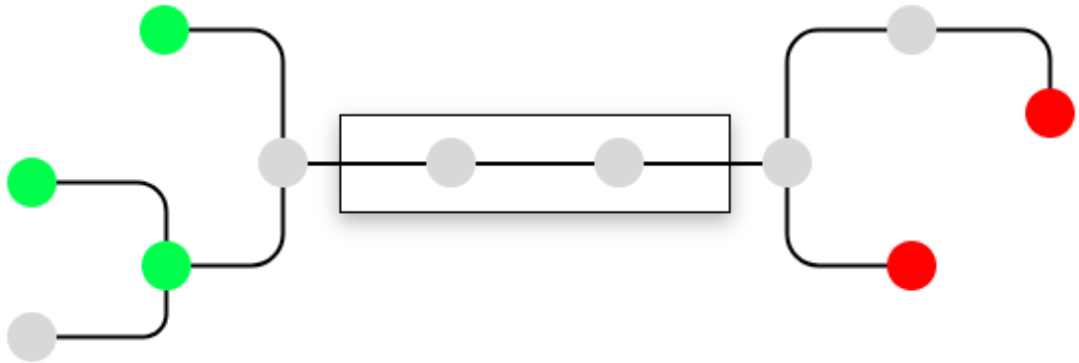
求好的有序对个数对 998244353 取模后的结果。

数据范围：$n \leq 2 \times 10^5$。

Problem E. 二分图问题

包含 S 的最小连通子图和包含 T 的最小连通子图不交，实际上等价于：存在一些点/边，使得在断掉点/边后，S 和 T 各自出现在两个不同的连通块内。

进一步研究发现，满足上述条件的点和边恰好形成了一条链，对应的是两个最小连通子图之间的最短路径。需要注意，这条路径上不包含两个端点。



Problem E. 二分图问题

考虑将计数问题放在链上。对每个点 v 记 f_v 表示删除点 v 后，在剩余的连通块内选择两个，分别产生 S 和 T 的方案数，对每条边 e 记 g_e 表示删除边 e 后，在剩余的两个连通块内分别产生 S 和 T 的方案数。不难发现，对于一个合法的有序对 (S,T)，其在整条链对应的 f_v/g_e 上均产生了 1 的贡献。考虑到链上的边数比点数恰好多 1，那么只需要求出如下表达式的值，就能将每个有序对的贡献变为 1，从而计算答案：

$$\sum_{e\in E} g_e - \sum_{v\in V} f_v$$

Problem E. 二分图问题

剩下的问题就是求解所有 f_v 和 g_e。实际上，这两个数字的值可以通过预处理如下两个数组后借助组合数学计算：inner_v，代表在 v 的子树内部选择一个颜色相同的点集的方案数，以及 outer_v，代表在 v 的子树外部选择一个颜色相同的点集的方案数。

记 $C(v,i)$ 表示 v 的子树内颜色等于 i 的点数，则这两个值可以通过对每个点处理如下两个数组计算：

$$\text{inner}_v = \sum_{c=1}^{n} \left(2^{C(v,c)} - 1\right), \text{outer}_v = \sum_{c=1}^{n} \left(2^{C(1,c)-C(v,c)} - 1\right)$$

上述表达式在改变其中一个颜色包含的点数时贡献都可以 $\mathcal{O}(1)$ 维护，因此使用 DSU on Tree 即可做到 $\mathcal{O}(n \log n)$ 的复杂度。

另外，也存在通过两个连通子图的交集上进行点边容斥计算对应的 $\mathcal{O}(n \log n)$ 做法，以及其余可能的计数方案，在此不展开讨论。

Problem H. 哈希

题意简述

定义 sz_x 为 x 的子树包含的点数，而 son_x 表示 x 的所有子节点形成的集合，对于一棵以 r 为根的有根树，定义其哈希值 $h(r)$ 为：

$$h(r) = 1 + \sum_{v \in \mathrm{son}_r} h(v) \times f(\mathrm{sz}_v)$$

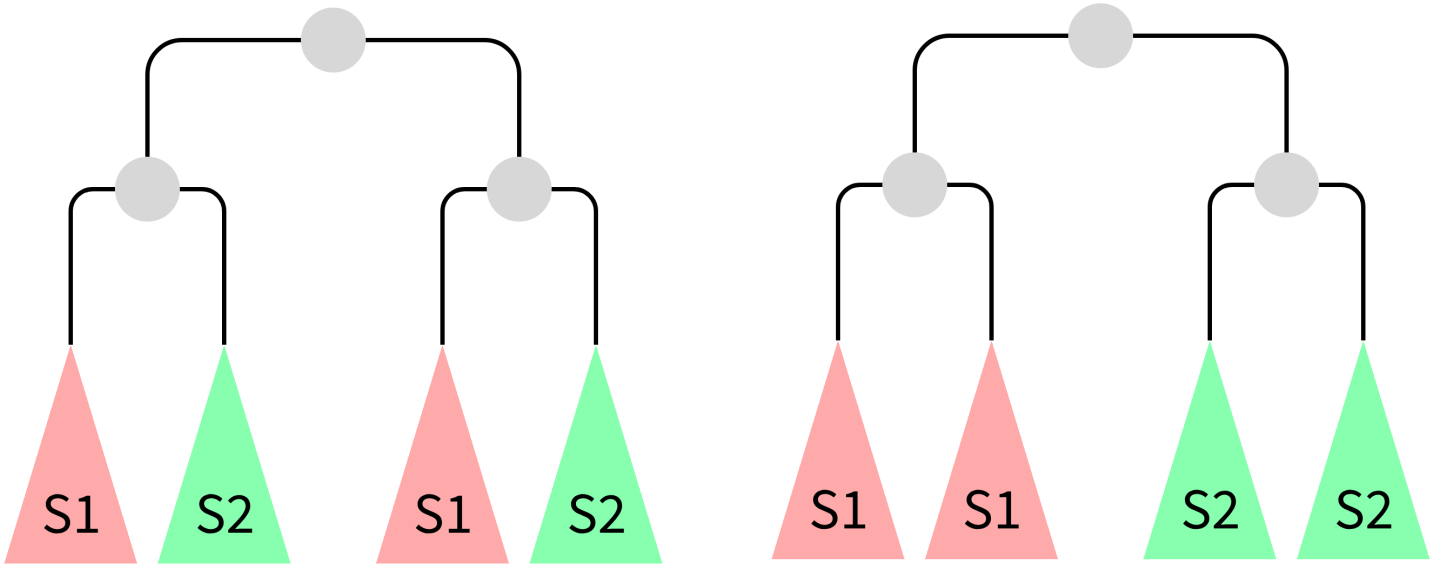
其中的 f 代表任意一个从正整数集合到正整数集合的映射，即 $f \in \{\varphi : \mathbb{Z}^+ \to \mathbb{Z}^+\}$。

对于两棵有根树，如果存在一个选取 f 的方案，使得二者能够计算出不同的哈希值，则称二者是可区分的。求出最多能选择多少个包含 n 个节点的有根二叉树，使得它们是两两可区分的。答案模数 p 通过输入给出。

数据范围：$n \le 3000, \sum n \le 10000$。

Problem H. 哈希

首先需要注意：两棵有根二叉树不可区分，并不意味着二者同构。以下是两个不可区分但不同构的有限二叉树，其中 S_1 和 S_2 代表两个大小相同但不同构的二叉树结构。



Problem H. 哈希

观察哈希值对应的表达式：

$$h(r) = 1 + \sum_{v \in \text{son}_r} h(v) \times f(\text{sz}_v)$$

那么对于每个点 v，自身哈希值中“1”部分的贡献会通过其父节点转移到根，中途会不断乘以祖先上所有点的大小在映射后对应的值。换句话说，对于节点 v，令它到根的路径上经过的所有点，除掉根本身对应的集合为 S_v，那么有

$$h(r) = \sum_{v} \prod_{w \in S_v} f(\text{sz}_w)$$

将所有点 v 对应的 S_v 放在一个可重集中，称为有根二叉树的标签。直觉告诉我们：通过寻找一个足够大的 f，就能对两个不同的标签产生不一样的哈希值。

Problem H. 哈希

一种可行的做法是：寻找一个足够大的整数 A，并且令 $B = 10^9$，定义 $f(x) = A^{B^x}$。

考虑项 $\prod_{w \in S_v} f(\mathrm{sz}_w)$，应用上述映射后得到 $A^{\sum_{w \in S_v} B^{\mathrm{sz}_w}}$。这里的 B 足够大，从 B 进制加法不会进位的角度看，不同的 S_v 必然会产生不同的 $\sum_{w \in S_v} B^{\mathrm{sz}_w}$。又考虑到 A 足够大，从 A 进制加法不会进位的角度看，不同签名得到的哈希值就是不同的。

另外可以发现，对于两个标签相同的有根二叉树，其哈希值必然相同，因此我们得到了结论：两棵有根二叉树可区分当且仅当二者的标签相同。

Problem H. 哈希

考虑动态规划，自然可以想到子树大小从 n 开始，不断分裂成两个子树，子树大小不断缩小直到退化为平凡情况。考虑到目前只有子树的大小是重要的，可以设立如下状态：$f_{i,j}$ 代表对于 j 个子树大小为 i 的点 v，在最后能得到几个两两可区分的结构。枚举其中一种分裂方法 $i = l + r + 1$，其中 $0 \le l \le r$，以及共有 k 个点遵循这一分拆方案，那么可以通过如下式子转移：

$$f_{i,j} \leftarrow f_{i,j-k} \times \begin{cases} f_{l,k} \times f_{r,k} & l \neq r \\ f_{l,2k} & l = r \end{cases}$$

从状态定义可以发现 $ij \le n$，即 $j \le \frac{n}{i}$。考虑“枚举 $i \to$ 枚举 $l \to$ 枚举 $j \to$ 枚举 k”的顺序，可以得到计算量不超过

$$\sum_{i=1}^{n} i \times \left(\frac{n}{i}\right)^2 = \sum_{i=1}^{n} \frac{n^2}{i} = \mathcal{O}(n^2 \log n)$$

Problem I. I Love CCPC

题意简述

定义一个 CCPC 串为满足“字符-字符-回文-字符”形式的字符串，其中要求三个“字符”相同。给定一个字符串 s，需要支持 n 次操作，每次操作会从左侧或者右侧插入一个新的字符，需要算出操作后字符串 s 的 CCPC 子串数量。

数据范围：$|s| \leq 5 \times 10^5$，$n \leq 5 \times 10^5$。

Problem I. I Love CCPC

将操作离线，问题可以等价为在某个大串 S 上求解 $[l,r]$ 部分子串的 CCPC 子串数，每次操作会将 l 左移一位或者将 r 右移一位。

自然想到增量法。在子串范围为 $[l,r]$ 的时候：

- 如果需要将 l 向左移动一位，那么新的 CCPC 子串需要以 $l-1$ 开头，继续考虑 CCPC 串的性质可以发现，子串 $[l-1,p]$ $(l+2 \le p \le r)$ 是 CCPC 子串，需要满足：$S_{l\ldots p}$ 是回文串，并且 $S_l = S_{l-1}$；
- 如果需要将 r 向右移动一位，那么新的 CCPC 子串需要以 $r+1$ 结尾。考虑到”CPC“本身也是一个回文串，那么子串 $[p,r+1]$ $(l \le p \le r-2)$ 是 CCPC 子串，需要满足：$S_{p+1\ldots r+1}$ 是回文串，并且 $S_p = S_{p+1}$。

Problem I. I Love CCPC

如果需要将 r 向右移动一位，那么新的 CCPC 子串需要以 $r+1$ 结尾。考虑到”CPC“本身也是一个回文串，那么子串 $[p, r+1]$ $(l \le p \le r-2)$ 是 CCPC 子串，需要满足：$S_{p+1\ldots r+1}$ 是回文串，并且 $S_p = S_{p+1}$。

考虑 $S_{1\ldots r+1}$ 的所有回文后缀的左端点 $[p_1, p_2, p_3, \ldots, p_k]$，假设这里有 $p_i < p_{i+1}$。对串 S 建立回文自动机，那么可以通过 fail 树上跳父节点得到所有的 p_i。另外，这些左端点还需要落在 $[l, r+2]$ 以内，可以使用倍增算法找到对应的部分。

随后可以发现，除了满足条件的最长回文后缀之外，其余后缀是否产生贡献实际上和 l 本身没有关系，因此它们一定在最长回文后缀的内部，其内容也是固定的。因此可以在 fail 树上处理贡献的链和，就能够在总体 $\mathcal{O}(\log|S|)$ 的时间复杂度内实现。

Problem I. I Love CCPC

如果需要将 l 向左移动一位，那么新的 CCPC 子串需要以 $l-1$ 开头，继续考虑 CCPC 串的性质可以发现，子串 $[l-1,p]$ $(l+2\le p\le r)$ 是 CCPC 子串，需要满足：$S_{l\dots p}$ 是回文串，并且 $S_l=S_{l-1}$。

可以发现，$S_l=S_{l-1}$ 这一限制条件和 p 无关，因此只需要数从 l 开始且长度在 $[3,r-l+1]$ 范围内的回文串个数即可。这一问题可以使用各种经典回文串相关算法实现。

使用正反串回文自动机加倍增可以做到总体 $\mathcal{O}(|S|\log|S|)$ 的复杂度。当然，本题的性质非常友好，允许使用 Border 理论分析或者前后插入回文自动机等方式通过，在此不再展开。

Problem J. 记忆，排列和有根树

题意简述

给定一棵有根树和一个排列。对树上每个节点 u，称其**子树内**一个节点 v 是重要的，当且仅当 $u \neq v$，并且在排列中，v 的出现位置在 u 的前面。令 d_u 为所有对 u 重要的节点和 u 的距离最小值，若没有则为 -1。

现在需要完成这道题的逆版本，给定有根树和 d_i 序列，求出**字典序最小**的排列使得通过上述方法可以算出对应的 d_i 序列，或者指出这样的排列不存在。

数据范围：$n \leq 5 \times 10^5$。

Problem J. 记忆，排列和有根树

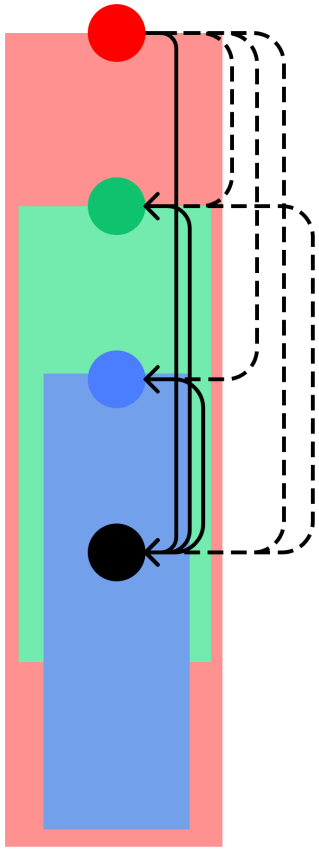
为方便讨论，将 $d_u = -1$ 更改为 $d_u = +\infty$。

首先，可以使用拓扑序刻画本题的限制。对于一个点 u，题目中的限制等价于：

- 对它子树内距离小于 d_u 的点，需要从点 u 连一条有向边；
- 在子树内距离等于 d_u 的点，需要选择任意一个点，向 u 连边。

先将所有第一类边连接，考虑所有有向边都是树上的祖先 - 后代边，那么只需要保证每条第二类边连上去不会形成环，那么所有第二类边同时连上去也不会形成环。

Problem J. 记忆，排列和有根树



首先考虑如何优化第一类边的连接。考虑到一个点，这个点需要被多个点相连，可以发现在其祖先的链上，这些点应当会形成类似祖先关系的形式。如左图所示，需要从红、绿、蓝三个点向黑点连边，但与此同时，红点也需要向蓝点和绿点连边，绿点也需要向蓝点连边。

这些祖先 - 后代边会形成一个类似完全图的形式，因此只需要连接最长链即可完成限制（在左图中，对应的是红点 $\to$ 绿点 $\to$ 蓝点 $\to$ 黑点的链）。这代表每个点只需要找到其祖先路径上最深且需要连边的点连边即可。可以使用倍增算法等找到需要连接的所有第一类边，对应的时间复杂度为 $\mathcal{O}(n \log n)$。

Problem J. 记忆，排列和有根树

对于第二类边，为了保证拓扑排序得到的字典序最小，无法在拓扑排序之前就连边。相反，需要根据拓扑排序的中间结果进行决策。

对所有 $d_u \neq +\infty$ 的点 u，称 V_u 表示其子树深度内和 u 距离恰为 d_u 的点集，另外定义 dep_u 代表点 u 的深度。对所有 $\mathrm{dep}_u + d_u$ 相同的点，可以证明它们对应的 V_u 的包含关系形成了树结构。这一树结构可以使用虚树或单调栈等方式得到。

在拓扑排序中，所有 V_u 非空的点都需要被**锁定**，也即不放在队列中。每次从队列取出最小的 x 时，对一些锁定的点 u，只要 V_u 中包含了 x，就可以将这些点 u **解锁**，加入到队列中。

在树结构中，包含了某个点 x 的所有集合形成了树上一个祖先 - 后代方向的链。实际上，只需要暴力从包含 x 的最小集合开始往上跳，直到到达一个已经被解锁的点对应的集合，就能得到所有需要解锁的点。

最后可以得到时间复杂度为 $\mathcal{O}(n \log n)$。

Problem B. 堡垒

题意简述

在圆内有 n 个矩形阻挡视线，求出从圆内一个观测点出发可以看到的圆内面积。

数据范围：$n \leq 10^5$。

Problem B. 堡垒

自然想到将面积拆分成若干个“扇形”，每个“扇形”单独考虑。具体地，从观测点向所有矩形的顶点发出射线，将会把圆周切除最多 $4n$ 段，在每一段上单独考虑。这里对扇形加引号，是因为观测点不一定在圆心。

在上述方式下，一个“扇形”内部会出现若干横线和若干竖线，并且这些线是贯穿整个“扇形”的，也即端点不严格落在“扇形”内部。首先可以发现，在所有横线中，只有离观测点最近的横线是需要考虑的，竖线同理。可以使用数据结构维护距离观测点最近的横线和竖线。

Problem B. 堡垒

接下来对“扇形”内部进行讨论：如果“扇形”内部没有线，则需要计算“扇形”的面积（可以使用三角形 + 弓的方式计算）；如果有一条线，则计算和观测点围成的三角形面积；否则，如果同时有横线和竖线，考虑二者所在直线的交点：

- 如果交点在“扇形”之外，说明二者在“扇形”内部没有交，分别计算两条线围成的面积后取较小值即可；
- 否则，二者在内部有交，可以发现可观测面积是一个四边形，同理计算即可。

最终的实现还需要注意一些细节问题，包括但不限于观测点在某个矩形的边或者顶点上等。最终的时间复杂度为 $\mathcal{O}(n \log n)$。

Problem D. 灯塔

题意简述

给定一棵包含 n 个点和 m 条无向边的边仙人掌，需要选出最少的点放置灯塔，每个灯塔可以找到距离不超过 k 的所有点，需要保证所有点都能被照到。

数据范围：$n, m \leq 2 \times 10^5$。

Problem D. 灯塔

在树上讨论这个问题时，我们的做法是：逐个子树进行贪心，求出子树内选择至少多少个灯塔就能满足条件，以及最好情况下从根向外还能向外照多远，或者需要从根往下照多远。在计算一个点的所有子树后，如果不在这个位置放置灯塔会导致一些点永远无法被照到，则必须在此处放一座灯塔，同时向内外照射，否则可以选择不放置灯塔。

考虑将这个做法放在仙人掌上。首先建立圆方树，通过圆方树确定环之间的树结构。接下来考虑其中一个环，其中某个点连向了圆方树上的祖先，是需要求解的值，而另外一些点则已经计算了其子树内部的答案。

Problem D. 灯塔

首先，环上一些点对应的情况是可以从里面向外照，这有可能会满足环上相邻节点向内照的限制，因此需要跑一次多源 01 最短路或动态规划，确认哪些点的限制已经被满足了。剩下无法被满足的限制可以表示为：在环的某一段上需要放置至少一座灯塔。

考虑断环为链，在数轴上考虑类似的问题时，假设区间可以表示为 $[l_i, r_i]$，那么在某个位置 x 放置灯塔之后，下一个需要放置的位置就是 $\min_{l_i>x} r_i$，也就是左端点超出 x 的所有区间中右端点的最小值。这一做法可以使用倍增优化，实现 $\mathcal{O}(n \log n)$ 预处理和 $\mathcal{O}(\log n)$ 查询从某个位置开始的选择方案。

将同样的思想放在环上，从祖先方向的点处将环断为链，在链上枚举第一座灯塔的位置，通过倍增方法可以对大小为 c 的环在 $\mathcal{O}(c \log c)$ 的复杂度内计算，在只把灯塔放在环内的前提下，至少需要多少座灯塔满足条件。

Problem D. 灯塔

上述灯塔需要全部在环内，而实际上，存在和树上问题一样的情况——可以选择一座灯塔放在祖先方向上，通过向内照的部分满足要求。此时需要考虑二分祖先方向的灯塔向内照的距离，在检测函数内部，可以根据需要往内照的距离，将已有的一些区间取消，随后通过同样的方法计算出最少需要放置多少座灯塔，如果数量小于仅在环内放置需要的数量，则认为可以选择这个距离，否则认为不能选择这个距离。

因此，本题可以在 $\mathcal{O}(n \log n \log k)$ 的时间复杂度内完成。

Problem M. MEX 问题

题意简述

对于一个长度为 n 的排列 p，定义 $f(p)$ 为满足 $i \in [1, n]$ 且 $[1, i]$ 范围内的 i 个数字在 p 中相连的 i 个数。

给定 n，对所有 $i \in [i, n]$ 求满足长度为 n 且 $f(p) = i$ 的排列 p 个数。对 998244353 取模。

数据范围：$n \leq 10^5$。

Problem M. MEX 问题

特判 $n = 1$ 的情况。

不妨将这一问题转换为如下钦定类型的题目，最后可以使用二项式反演求出答案。

对于一个排列 p，如果已经确认 $l_0 = 1, l_1, l_2, ..., l_{i-1} = n$ 是满足条件的 i，对应的方案数需要如何计算？实际上，对所有 $1 \le j < i$，需要将 $[l_{j-1} + 1, l_j]$ 范围内的所有整数放置在长度为 l_{j-1} 的排列两侧即可。令 $d_i = l_i - l_{i-1}$，那么这里就需要对 d_i 个数字进行随机排列，然后选择任意一个分割点将排列分开，放在左右两侧。可以得到对应的方案数是 $(d_i + 1)!$。

换言之，钦定了有 i 个值满足题述条件对应的方案数就是：

$$\sum_{x_0+x_1+...+x_{i-1}=n-1} \prod_{0 \le j < i} (x_j + 1)!$$

令 $F(u) = \sum_{i>0} (i+1)! u^i$，那么需要计算的值就是 $[u^{n-1}]F(u)^i$。

Problem M. MEX 问题

一种比较简单粗暴的方式是套用 Bostan-Mori，不过这一做法较新且常数略大，这里提供另一种常数较小的思路。

令 $H(u,x)=\sum_{i\geq 0}u^i x^i$，那么问题可以变成计算 $[u^{n-1}]H(F(u),x)$ 对应的多项式。利用拉格朗日反演，有

$$[u^{n-1}]H(F)=\frac{1}{n-1}[x^{n-2}]H'\left(\frac{x}{G}\right)^{n-1}$$

其中 G 是 F 的复合逆，也就是满足 $F(G(u))\equiv u$ 的多项式。考虑到 F 具有很好的形式，不妨考虑寻找一个较快的方式计算这个复合逆。

Problem M. MEX 问题

有 $F(u)=\sum_{i>0}(i+1)!u^i$，求导后得到 $F'(u)=\sum_{i\geq 2} i!\times(i-1)u^{i-2}$。考虑到 $(i+1)!=i!\times((i-1)+2)$，可以推导出 $F=u^2F'+2uF+2u$，也就是 $F'=\frac{F-2uF-2u}{u^2}$。那么：

$$F(G(u))=u$$

$$F'(G(u))G'(u)=1$$

$$\left(\frac{u-2uG(u)-2G(u)}{G^2(u)}\right)G'(u)=1$$

$$uG(u)-2uG(u)G'(u)-2G(u)G'(u)=G^2(u)$$

Problem M. MEX 问题

$$uG(u) - 2uG(u)G'(u) - 2G(u)G'(u) = G^2(u)$$

考虑左右两侧 u^n 的系数：

$$nG_n - 2\sum_{i=1}^{n} iG_iG_{n-i} - 2\sum_{i=1}^{n} iG_iG_{n-i+1} = \sum_{i=0}^{n} G_iG_{n-i}$$

进一步注意到 $G_0 = 0, G_1 = \frac{1}{2}$，代入整理得到

$$G_n = -\sum_{i=1}^{n-1} G_iG_{n-i} - 2\sum_{i=1}^{n-1} iG_iG_{n-i} - 2\sum_{i=2}^{n-1} iG_iG_{n-i+1}$$

使用半在线卷积或分治 NTT 即可完成求解，时间复杂度为 $\mathcal{O}(n \log^2 n)$。

Problem G. 贵校是构造王国吗 IV

题意简述

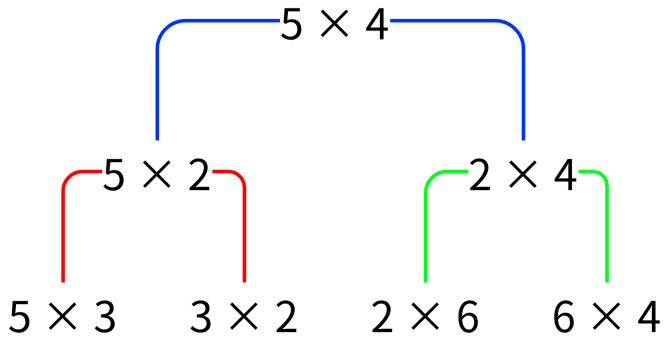
定义 $\mathrm{MCOP}([w_1, w_2, ..., w_n])$ 为对 $w_1 \times w_2, w_2 \times w_3, ..., w_{n-1} \times w_n$ 大小的矩阵进行矩阵链乘积所需的最少总运算数。

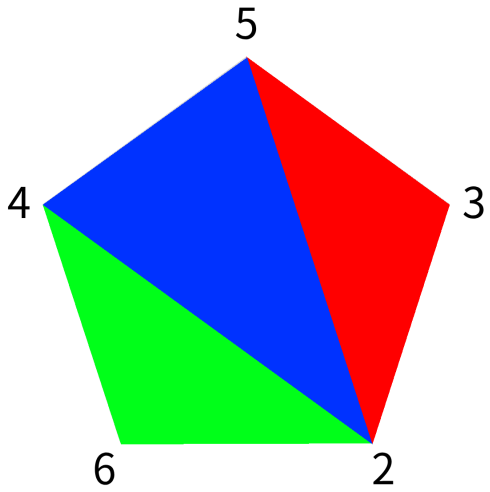
T 次询问，每次询问给定一个整数 X，需要构造一个长度 l 不超过 6 并且元素不超过 4000 的正整数序列 a，使得 $\mathrm{MCOP}(a) = X$。

数据范围：$T \leq 500, 1 \leq X \leq 10^9$。

Problem G. 贵校是构造王国吗 IV

在讨论这一问题前，需要将 MCOP 问题转化为更容易分析的形式。实际上，MCOP($[w_1, w_2, ..., w_n]$) 等价于对带点权凸多边形 $w_1 - w_2 - ... - w_n - w_1$ 求最小带权三角剖分，其中三角剖分的权值定义为所有三角形三个顶点权值乘积的和。

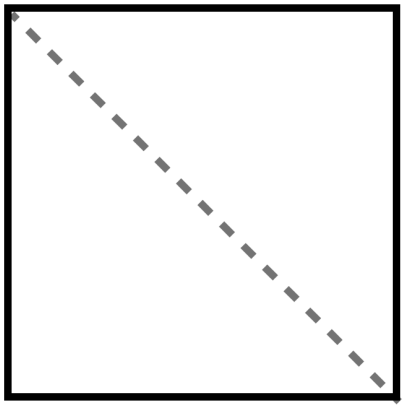


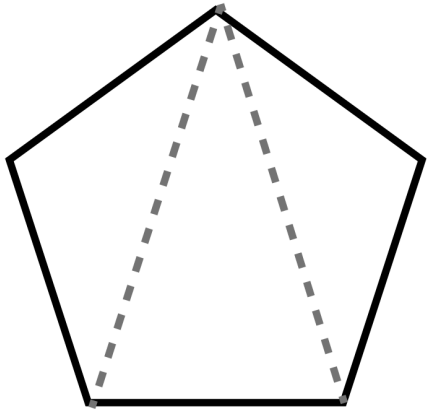


Problem G. 贵校是构造王国吗 IV

本道题需要回答两个问题：为什么 $l = 6$，以及如何 $l = 6$?

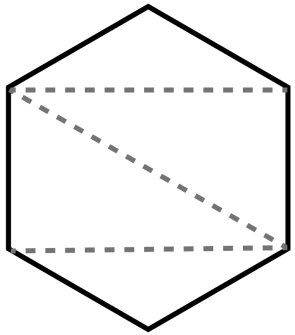
对于为什么 $l = 6$ 的问题，需要注意到：四边形和五边形的所有三角剖分中，都会有一个点向其余所有点连边。此时可以发现，三角剖分的权值一定是这一个点对应权值的倍数，那么只需要将 X 设置为一个足够大的质数，就能保证 $l \leq 5$ 时无法构造出对应解。

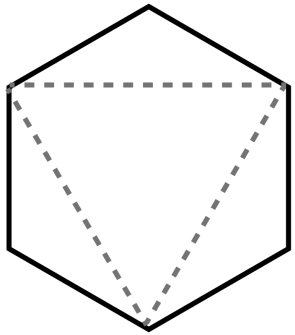




Problem G. 贵校是构造王国吗 IV

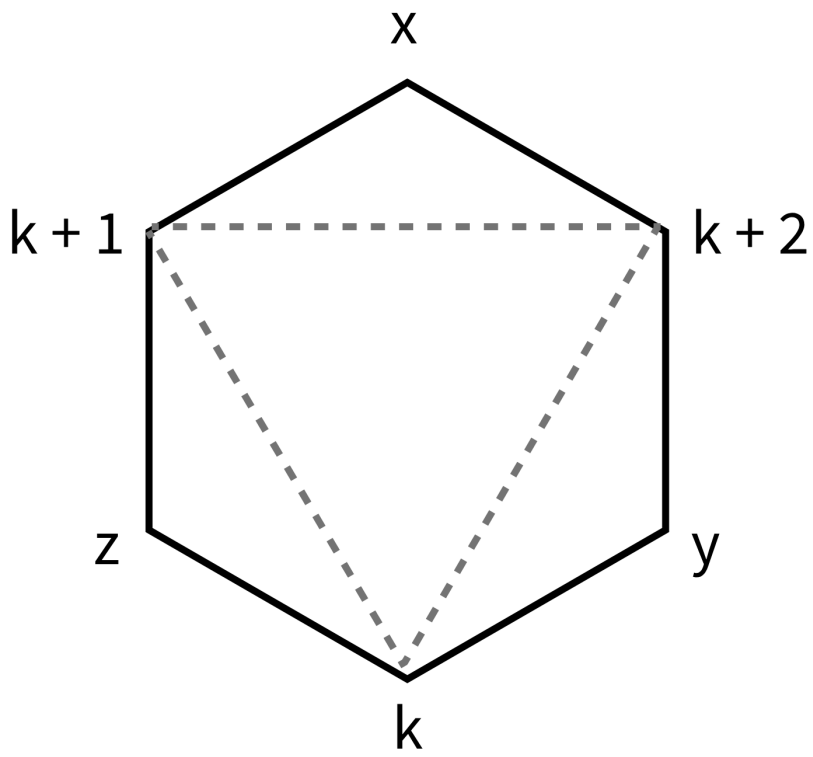
接下来考虑如何 $l = 6$。注意到六边形的三角剖分中，为了使得没有点连接了剩余五个点，只有两种本质不同的剖分方案：





正解利用的是右侧的剖分。通过归纳法可以证明：必然存在一种剖分方法，使得点权最小的点向点权第二小和第三小的点连边。因此在右侧的剖分中，将较小的三个值填写在中间三角形的部分是比较有利的。

Problem G. 贵校是构造王国吗 IV



一种比较直观的方式是将中间三角形的三个点权设置为 $k, k+1, k+2$，其中 k 是一个**奇数**（若是偶数，则剖分权值一定是偶数），如左图所示。将剩余三个点的点权设为 x, y, z。为了保证 $k, k+1, k+2$ 是最小的三个点，需要保证 $x, y, z \geq k+2$。

y 和 z 不需要满足额外限制，而 x 需要满足：将 $k+1$ 和 $k+2$ 相连，比将 x 和 k 相连更优。可以得到如下限制：$\frac{1}{k} + \frac{1}{x} < \frac{1}{k+1} + \frac{1}{k+2}$。不难想象，此时 x 相较于 k 不会很大。

Problem G. 贵校是构造王国吗 IV

在得到 x, y, z 的最小值之后，可以将三个数字分别加上一个自然数，在不超过 4000 的情况下凑出所需的 X。此时问题转换为：有三个物品，价值分别为 $k(k+1), k(k+2), (k+1)(k+2)$，需要凑出某个总价值。

首先考虑 $k(k+1)$ 和 $(k+1)(k+2)$，可以凑出 $k+1$ 的倍数。在除去 $k+1$ 之后，需要注意到 $\gcd(k, k+2) = 1$，因此根据 Frobenius 定理可以知道，通过这两个数可以保证对所有不小于 $k(k+2) - k - (k+2) + 1$ 的数字，总能凑出这个数的 $k+1$ 倍。

然后通过 $k(k+2)$ 调整总价值对 $k+1$ 取模后的结果，可以发现在不超过 k 次调整后即可满足所有模 $k+1$ 的可能性。因此为了保证所有数都能凑出，一个下界是

$$(k+1)[k(k+2) - k - (k+2) + 1] + k^2(k+2)$$

Problem G. 贵校是构造王国吗 IV

最后，对每个奇数 k，首先统计默认情况（$y = z = k + 2$，x 是满足 $\frac{1}{k} + \frac{1}{x} < \frac{1}{k+1} + \frac{1}{k+2}$ 的最小整数）对应的结果 T，随后设置这一构造任意可行的下界为

$$T + (k+1)[k(k+2) - k - (k+2) + 1] + k^2(k+2)$$

上界为

$$T + \min(\{(k+1)(k+2)(4000 - x), k(k+2)(4000 - y), k(k+1)(4000 - z)\})$$

在预处理 $k \leq 1000$ 的所有区间后，发现其覆盖了从 $[A, 10^9]$ 范围内的所有数字，其中 A 对应的数字足够小，可以作为特殊情况处理。为了构造方案，需要对每次询问处理 Frobenius 问题的构造，因此最终复杂度可以做到 $\mathcal{O}(T1000^2)$ 以内。

事实上，正解做法可以实现 $a_i \leq 2000$ 的限制，为了保证其他类似的做法可以通过，启用了两倍的限制。