第十七届吉林省省赛简要题解

吉林省赛出题组

2024 年 5 月 18 日

前言

- 预计难度：$\mathsf{I} < \mathsf{GL} < \mathsf{F} < \mathsf{BDE} < \mathsf{CH} < \mathsf{AJK}$。
- 最终难度：ILGEBCDFAKHJ。

I. The Easiest Problem

简述题意

询问字符串"Scan the QR code to sign in now."有多少个小写字母。

I. The Easiest Problem

简述题意

询问字符串 “Scan the QR code to sign in now.” 有多少个小写字母。

总共有 21 个，输出 21 即可。

L. Recharge

简述题意

有无限个体积无限的背包，以及 x 个体积为 1,y 个体积为 2 的物品。请问如何将物品装到背包内，才能使装了至少体积为 k 的物品的背包尽量多？输出这个背包数量。

共 T 组询问。

数据范围：$k, x, y \leq 10^9, T \leq 2 \times 10^5$。

L. Recharge

简述题意

有无限个体积无限的背包，以及 x 个体积为 1,y 个体积为 2 的物品。请问如何将物品装到背包内，才能使装了至少体积为 k 的物品的背包尽量多？输出这个背包数量。

共 T 组询问。

数据范围：$k,x,y \leq 10^9, T \leq 2 \times 10^5$。

显然，答案存在一个上界 $\lfloor \frac{x+2y}{k} \rfloor$。

L. Recharge

简述题意

有无限个体积无限的背包，以及 x 个体积为 1,y 个体积为 2 的物品。请问如何将物品装到背包内，才能使装了至少体积为 k 的物品的背包尽量多？输出这个背包数量。

共 T 组询问。

数据范围：$k,x,y \le 10^9, T \le 2\times 10^5$。

显然，答案存在一个上界 $\lfloor \frac{x+2y}{k} \rfloor$。

在 $2 \mid k$ 时，可以通过先装 2 后装 1 的策略达到这个上界。

L. Recharge

在 $2 \nmid k$ 时，可以考虑二分答案。在二分答案后对于同一个背包尽量先放 2 背包满了/2 没了时再放 1 来判断是否合法。单组询问时间复杂度 $O(\log(x+y))$。

L. Recharge

在 $2 \nmid k$ 时，可以考虑二分答案。在二分答案后对于同一个背包尽量先放 2 背包满了/2 没了时再放 1 来判断是否合法。单组询问时间复杂度 $O(\log(x+y))$。

Bonus：考虑在单组询问 $O(1)$ 的时间复杂度内解决该问题。

D. Parallel Lines

简要题意

给定二维平面上的 n 个点以及 k 条未知的平行线。请你找到这 k 条直线，使得每个点都在某条直线上，且每条直线至少有两个点。

数据范围：$n \le 10^4, k \le 50$。

D. Parallel Lines

简要题意

给定二维平面上的 n 个点以及 k 条未知的平行线。请你找到这 k 条直线，使得每个点都在某条直线上，且每条直线至少有两个点。

数据范围： $n \leq 10^4, k \leq 50$。

对于一个固定的斜率 K，可以通过 $O(n \log n)$ 扫描判断其是否满足条件。而由于每条直线至少有两个点，所以合法的斜率 K 一定是某点 i 与某点 j 确定的直线的斜率。

D. Parallel Lines

简要题意

给定二维平面上的 n 个点以及 k 条未知的平行线。请你找到这 k 条直线，使得每个点都在某条直线上，且每条直线至少有两个点。

数据范围：$n \leq 10^4, k \leq 50$。

对于一个固定的斜率 K，可以通过 $O(n \log n)$ 扫描判断其是否满足条件。而由于每条直线至少有两个点，所以合法的斜率 K 一定是某点 i 与某点 j 确定的直线的斜率。

可以考虑直接随机 i 和 j 并判断，可以证明随机出的合法的 i, j 最小概率为 $\frac{1}{k}$，所以可以在 $O(k)$ 次随机找出满足条件的斜率 K 以及点的划分方式。时间复杂度 $O(nk \log n)$。

D. Parallel Lines

证明

显然，概率最小时 n 个点平均分配在 k 条直线上。不妨假设 $n = mk$，则随机两点 i, j 其在同一条直线的概率为：

$$\frac{k\binom{m}{2}}{\binom{n}{2}} = \frac{n(m-1)/2}{n(n-1)/2} = O(\frac{1}{k})$$

E. Connected Components

简述题意

给定一个 n 个点的无向图以及参数数组 $a_1, a_2, \ldots, a_n$ 与 $b_1, b_2, \ldots, b_n$。i 与 j $(i < j)$ 有边当且仅当 $a_i - a_j \le i - j \le b_i - b_j$ 或 $a_j - a_i \le j - i \le b_j - b_i$。
询问本图有多少个连通块。

数据范围：$n \le 10^6, 10^{-9} \le a_i, b_i \le 10^9$。

E. Connected Components

简述题意

给定一个 n 个点的无向图以及参数数组 $a_1, a_2, \ldots, a_n$ 与 $b_1, b_2, \ldots, b_n$。i 与 j $(i < j)$ 有边当且仅当 $a_i - a_j \le i - j \le b_i - b_j$ 或 $a_j - a_i \le j - i \le b_j - b_i$。
询问本图有多少个连通块。

数据范围：$n \le 10^6, 10^{-9} \le a_i, b_i \le 10^9$。

标准处理：对不等式变形，使得一边只与 i 有关，另一边只与 j 有关：

$$a_i - i \le a_j - j \vee i - b_i \le j - b_j$$

E. Connected Components

简述题意

给定一个 n 个点的无向图以及参数数组 $a_1, a_2, \ldots, a_n$ 与 $b_1, b_2, \ldots, b_n$。i 与 j $(i < j)$ 有边当且仅当 $a_i - a_j \le i - j \le b_i - b_j$ 或 $a_j - a_i \le j - i \le b_j - b_i$。
询问本图有多少个连通块。

数据范围：$n \le 10^6, 10^{-9} \le a_i, b_i \le 10^9$。

标准处理：对不等式变形，使得一边只与 i 有关，另一边只与 j 有关：

$$a_i - i \le a_j - j \vee i - b_i \le j - b_j$$

所以令 $x_i = a_i - i, y_i = i - b_i$，$i$ 与 j 有边的条件当且仅当 $x_i \le x_j \vee y_i \le y_j$。

E. Connected Components

考虑将所有点按 x 排序，再考虑按标号从小到大加入每个点以及更新连通情况。

E. Connected Components

考虑将所有点按 x 排序，再考虑按标号从小到大加入每个点以及更新连通情况。

在加入点 i 前，我们只关心每个连通块 y 最小的点，将这些点用一个单调栈维护。加入 i 时如果 y_i 小于栈顶则将 i 推入栈，否则将所有 $\leq y_i$ 的非栈顶元素弹出（表示 i 与这些点连边）并保留栈顶。

E. Connected Components

考虑将所有点按 x 排序，再考虑按标号从小到大加入每个点以及更新连通情况。

在加入点 i 前，我们只关心每个连通块 y 最小的点，将这些点用一个单调栈维护。加入 i 时如果 y_i 小于栈顶则将 i 推入栈，否则将所有 $\leq y_i$ 的非栈顶元素弹出（表示 i 与这些点连边）并保留栈顶。

在过程结束后栈中剩下的点的数量即为答案，时间复杂度 $O(n \log n)$。

G. Platform Game

简述题意

在二维平面上给定 n 条水平的线段表示平台，机器人在空中会垂直下落，在平台上会向右移动。

现在一个机器人从 (sx, sy) 出发，询问机器人在落地（到达 $y = 0$）时的 x 坐标。

数据范围：$n \leq 2 \times 10^5$

G. Platform Game

简述题意

在二维平面上给定 n 条水平的线段表示平台，机器人在空中会垂直下落，在平台上会向右移动。

现在一个机器人从 (sx, sy) 出发，询问机器人在落地（到达 $y = 0$）时的 x 坐标。

数据范围：$n \leq 2 \times 10^5$

模拟机器人的移动路径，可以发现机器人依次经过的平台 y 坐标严格递减。

G. Platform Game

简述题意

在二维平面上给定 n 条水平的线段表示平台，机器人在空中会垂直下落，在平台上会向右移动。

现在一个机器人从 (sx, sy) 出发，询问机器人在落地（到达 $y = 0$）时的 x 坐标。

数据范围：$n \le 2 \times 10^5$

模拟机器人的移动路径，可以发现机器人依次经过的平台 y 坐标严格递减。

按照 y 递减的顺序考虑平台，依次判断机器人能否落在该平台上：若能，则将机器人的坐标更新为其离开平台瞬间的坐标（即平台右端点的坐标）。

G. Platform Game

简述题意

在二维平面上给定 n 条水平的线段表示平台，机器人在空中会垂直下落，在平台上会向右移动。

现在一个机器人从 (sx, sy) 出发，询问机器人在落地（到达 $y = 0$）时的 x 坐标。

数据范围：$n \le 2 \times 10^5$

模拟机器人的移动路径，可以发现机器人依次经过的平台 y 坐标严格递减。

按照 y 递减的顺序考虑平台，依次判断机器人能否落在该平台上：若能，则将机器人的坐标更新为其离开平台瞬间的坐标（即平台右端点的坐标）。

最终机器人的 x 坐标即为答案。时间复杂度：$O(n \log n)$。

F. Best Player

简述题意

给定 n 位玩家以及 m 场决斗，第 i 场决斗由第 a_i 和 b_i 名玩家进行，他们分别会获得 $x_i + p_i$ 和 $y_i + q_i$ 的分数。其中 x_i, y_i 给定，$p_i + q_i = z_i$，z_i 给定，p_i, q_i 是可以自由选择的非负常数。

一名玩家获得的最终分数是他在所有决斗中单场得分的最大值，如果一名玩家的最终分数在所有玩家中严格最大，则他是赢家。请问哪些玩家可能成为赢家？

数据范围：$n, m \le 2 \times 10^5$。

F. Best Player

简述题意

给定 n 位玩家以及 m 场决斗，第 i 场决斗由第 a_i 和 b_i 名玩家进行，他们分别会获得 $x_i + p_i$ 和 $y_i + q_i$ 的分数。其中 x_i, y_i 给定，$p_i + q_i = z_i$，z_i 给定，p_i, q_i 是可以自由选择的非负常数。

一名玩家获得的最终分数是他在所有决斗中单场得分的最大值，如果一名玩家的最终分数在所有玩家中严格最大，则他是赢家。请问哪些玩家可能成为赢家？

数据范围：$n, m \leq 2 \times 10^5$。

如果 i 想成为赢家，最优情况应该是有 i 参与的决斗分数全部分配给 i，否则平均分配。

F. Best Player

直接暴力模拟该过程可以在 $O(m)$ 的时间复杂度内判断一名玩家，但是 $O(nm)$ 的时间复杂度仍然难以通过。

F. Best Player

直接暴力模拟该过程可以在 $O(m)$ 的时间复杂度内判断一名玩家，但是 $O(nm)$ 的时间复杂度仍然难以通过。

考虑优化，判断 i 时只枚举 i 参与的比赛，其他比赛可以用一个 STL（multiset 或可删堆）维护其他选手的得分最大值。最后时间复杂度 $O(n + m\log m)$。

B. Dfs Order 0.5

简述题意

给定一个大小为 n 的树，求出这棵树的一个 dfs 序 $dfn_1, dfn_2, \ldots, dfn_n$，使得最大化：

$$\sum_{i=1}^{n} [2 \mid dfn_i] a_i$$

数据范围：$n \le 2 \times 10^5$

B. Dfs Order 0.5

简述题意

给定一个大小为 n 的树，求出这棵树的一个 dfs 序 $dfn_1, dfn_2, \ldots, dfn_n$，使得最大化：

$$\sum_{i=1}^{n}[2 \mid dfn_i]a_i$$

数据范围：$n \leq 2 \times 10^5$

考虑 DP，假设 $dp_{u,0}$ 表示 u 子树内偶数 dfs 序 a 之和的最大值，$dp_{u,1}$ 表示 u 子树内奇数 dfs 序 a 之和的最大值。

B. Dfs Order 0.5

考虑如何转移，假设子树大小为偶数的点为**偶节点**，子树大小为奇数的点为**奇节点**，可以按照以下两种情况考虑：

B. Dfs Order 0.5

考虑如何转移，假设子树大小为偶数的点为**偶节点**，子树大小为奇数的点为**奇节点**，可以按照以下两种情况考虑：

B. Dfs Order 0.5

考虑如何转移，假设子树大小为偶数的点为**偶节点**，子树大小为奇数的点为**奇节点**，可以按照以下两种情况考虑：

- u 的所有儿子都是偶节点：
  此时遍历顺序不影响转移，有转移式子：
  $$dp_{u,t} = \sum_{v \in son(u)} dp_{v,t\oplus 1} + t \times a_u \quad t \in \{0,1\}$$

B. Dfs Order 0.5

考虑如何转移，假设子树大小为偶数的点为**偶节点**，子树大小为奇数的点为**奇节点**，可以按照以下两种情况考虑：

- u 的所有儿子都是偶节点：
  此时遍历顺序不影响转移，有转移式子：
  $$dp_{u,t} = \sum_{v \in son(u)} dp_{v,t \oplus 1} + t \times a_u \quad t \in \{0, 1\}$$
- u 的儿子存在奇节点：
  对于偶节点，其可以任意选择 $0/1$，故将 $dp_{v,0}/dp_{v,1}$ 的最大值贡献到 $dp_{u,t}$ 即可。
  对于奇节点，其一半取 0，一半取 1，如果有奇数个则还有一个取 $t \oplus 1$。所以按照 $dp_{u,0} - dp_{u,1}$ 的大小贪心即可。

B. Dfs Order 0.5

考虑如何转移，假设子树大小为偶数的点为**偶节点**，子树大小为奇数的点为**奇节点**，可以按照以下两种情况考虑：

- u 的所有儿子都是偶节点：
  此时遍历顺序不影响转移，有转移式子：
  $$dp_{u,t} = \sum_{v \in son(u)} dp_{v,t\oplus 1} + t \times a_u \quad t \in \{0,1\}$$
- u 的儿子存在奇节点：
  对于偶节点，其可以任意选择 $0/1$，故将 $dp_{v,0}/dp_{v,1}$ 的最大值贡献到 $dp_{u,t}$ 即可。
  对于奇节点，其一半取 0，一半取 1，如果有奇数个则还有一个取 $t \oplus 1$。所以按照 $dp_{u,0} - dp_{u,1}$ 的大小贪心即可。

故完成该树形 DP 后，$dp_{1,0}$ 即为答案。时间复杂度 $O(n \log n)$。

C. Fibonacci Sum

简述题意

假设 $g(x)$ 表示 x 二进制表示 1 的数量，$f(n)$ 表示斐波那契数列第 n 项。给定 n，求：

$$\sum_{i=1}^{n} f(g(i))$$

数据范围：$n \le 2^{10^7}$。

C. Fibonacci Sum

可以考虑将求 g 的过程写作矩阵乘法的过程，令：

$$T_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, T_1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, F_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}, T = T_0 + T_1$$

C. Fibonacci Sum

可以考虑将求 g 的过程写作矩阵乘法的过程，令：

$$T_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, T_1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, F_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}, T = T_0 + T_1$$

例：

$$f(g(101_2)) = F_0 \cdot T_1 \cdot T_0 \cdot T_1$$

C. Fibonacci Sum

可以考虑将求 g 的过程写作矩阵乘法的过程，令：

$$T_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, T_1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, F_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}, T = T_0 + T_1$$

例：

$$f(g(101_2)) = F_0 \cdot T_1 \cdot T_0 \cdot T_1$$

所以，有：

$$\sum_{i=0}^{2^m-1} f(g(i)) = F_0 T^m$$

C. Fibonacci Sum

由于 $0, 1, 2, \ldots, n$ 可以拆解成 $O(\log n)$ 个固定前缀 + 任意后缀的区间。(例如 $110 = 0xx + 10x + 110$)。固定的前缀贡献为 T_0 或 T_1，任意的后缀贡献为 T。

C. Fibonacci Sum

由于 $0, 1, 2, \ldots, n$ 可以拆解成 $O(\log n)$ 个固定前缀 + 任意后缀的区间。(例如 $110 = 0xx + 10x + 110$)。固定的前缀贡献为 T_0 或 T_1，任意的后缀贡献为 T。

所以答案可以拆解成 $O(\log n)$ 个若干个 T_0, T_1 与 T 的乘积。可以考虑预处理 T^i 来快速求得答案。

C. Fibonacci Sum

由于 $0, 1, 2, \ldots, n$ 可以拆解成 $O(\log n)$ 个固定前缀 + 任意后缀的区间。(例如 $110 = 0xx + 10x + 110$)。固定的前缀贡献为 T_0 或 T_1，任意的后缀贡献为 T。

所以答案可以拆解成 $O(\log n)$ 个若干个 T_0, T_1 与 T 的乘积。可以考虑预处理 T^i 来快速求得答案。

时间复杂度 $O(2^3 \log n)$。

H. Games on the Ads 2: Painting

简要题意

给定一个 $n \times n$ 的目标染色矩阵以及 n 行与 n 列的染色刷子，保证行和列的刷子颜色都是一个 n 的排列，询问 $2n!$ 种方案中有多少种染色方案能达成目标染色矩阵。

数据范围：$n \le 20$。

H. Games on the Ads 2: Painting

简要题意

给定一个 $n \times n$ 的目标染色矩阵以及 n 行与 n 列的染色刷子，保证行和列的刷子颜色都是一个 n 的排列，询问 $2n!$ 种方案中有多少种染色方案能达成目标染色矩阵。

数据范围：$n \leq 20$。

对于一个格子 (i, j)，我们可以知道以下信息：

H. Games on the Ads 2: Painting

简要题意

给定一个 $n \times n$ 的目标染色矩阵以及 n 行与 n 列的染色刷子，保证行和列的刷子颜色都是一个 n 的排列，询问 $2n!$ 种方案中有多少种染色方案能达成目标染色矩阵。

数据范围：$n \leq 20$。

对于一个格子 (i, j)，我们可以知道以下信息：

- 如果 $c_{i,j} = p_i = q_j$，则没有信息。

H. Games on the Ads 2: Painting

简要题意

给定一个 $n \times n$ 的目标染色矩阵以及 n 行与 n 列的染色刷子，保证行和列的刷子颜色都是一个 n 的排列，询问 $2n!$ 种方案中有多少种染色方案能达成目标染色矩阵。

数据范围：$n \le 20$。

对于一个格子 (i, j)，我们可以知道以下信息：

- 如果 $c_{i,j} = p_i = q_j$，则没有信息。
- 如果 $c_{i,j} \ne p_i, c_{i,j} \ne q_j$，则问题无解。

H. Games on the Ads 2: Painting

简要题意

给定一个 $n \times n$ 的目标染色矩阵以及 n 行与 n 列的染色刷子，保证行和列的刷子颜色都是一个 n 的排列，询问 $2n!$ 种方案中有多少种染色方案能达成目标染色矩阵。

数据范围：$n \leq 20$。

对于一个格子 (i, j)，我们可以知道以下信息：

- 如果 $c_{i,j} = p_i = q_j$，则没有信息。
- 如果 $c_{i,j} \neq p_i, c_{i,j} \neq q_j$，则问题无解。
- 如果 $c_{i,j} = p_i \neq q_j$，则第 i 行的刷子后于第 j 列。

H. Games on the Ads 2: Painting

简要题意

给定一个 $n \times n$ 的目标染色矩阵以及 n 行与 n 列的染色刷子，保证行和列的刷子颜色都是一个 n 的排列，询问 $2n!$ 种方案中有多少种染色方案能达成目标染色矩阵。

数据范围：$n \le 20$。

对于一个格子 (i, j)，我们可以知道以下信息：

- 如果 $c_{i,j} = p_i = q_j$，则没有信息。
- 如果 $c_{i,j} \neq p_i, c_{i,j} \neq q_j$，则问题无解。
- 如果 $c_{i,j} = p_i \neq q_j$，则第 i 行的刷子后于第 j 列。
- 如果 $c_{i,j} = q_j \neq p_i$，则第 j 列的刷子后于第 i 行。

H. Games on the Ads 2: Painting

因此，如果问题有解，则问题转变为了一个 $n+n$ 个点，$n \times n - n$ 条边的有向无环图，求该图的拓扑序数量。

H. Games on the Ads 2: Painting

因此，如果问题有解，则问题转变为了一个 $n+n$ 个点，$n\times n-n$ 条边的有向无环图，求该图的拓扑序数量。

由于拓扑序计数是 NP 问题，如果对这个问题直接计数，时间复杂度为 $O(2^{2n}\times poly(n))$，难以通过。

H. Games on the Ads 2: Painting

因此，如果问题有解，则问题转变为了一个 $n+n$ 个点，$n \times n - n$ 条边的有向无环图，求该图的拓扑序数量。

由于拓扑序计数是 NP 问题，如果对这个问题直接计数，时间复杂度为 $O(2^{2n} \times poly(n))$，难以通过。

假设这是一个 $n+n$ 个点，$n \times n$ 条边的有向无环图，可以发现，该图一定满足（假设左边点集为 S，右边点集为 T）：

在拓扑排序的某个时刻，要么只有 S 中的点入度为 0，要么只有 T 中的点入度为 0。

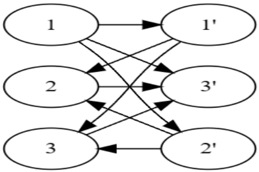
H. Games on the Ads 2: Painting

因此可以发现，该图的拓扑序可以分割成若干个点集，每个点集内的顺序可以任意排列，点集与点集之间的顺序不能改变。所以答案应该为所有点集大小的阶乘之积。

H. Games on the Ads 2: Painting

因此可以发现，该图的拓扑序可以分割成若干个点集，每个点集内的顺序可以任意排列，点集与点集之间的顺序不能改变。所以答案应该为所有点集大小的阶乘之积。

一个例子：

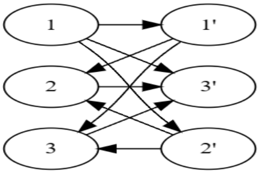


该图的拓扑序为 $\{1\},\{1',2'\},\{2,3\},\{3\}$，答案为 $1!\cdot 2!\cdot 2!\cdot 1!=4$ 种。

H. Games on the Ads 2: Painting

因此可以发现，该图的拓扑序可以分割成若干个点集，每个点集内的顺序可以任意排列，点集与点集之间的顺序不能改变。所以答案应该为所有点集大小的阶乘之积。

一个例子：



该图的拓扑序为 $\{1\},\{1',2'\},\{2,3\},\{3\}$，答案为 $1!\cdot 2!\cdot 2!\cdot 1! = 4$ 种。

因此对于缺失的 n 条边，我们直接枚举所有 2^n 种可能的情况，然后按照 n^2 条边的思路进行拓扑排序即可。时间复杂度 $O(2^n n^2)$。

A.Eminor Array

简述题意

询问有多少值在 1 到 2^n-1 的严格递增序列，使得不存在相邻的三个数异或和为 0。

数据范围：$n \le 10^6$。

A.Eminor Array

简述题意

询问有多少值在 1 到 $2^n - 1$ 的严格递增序列，使得不存在相邻的三个数异或和为 0。

数据范围：$n \le 10^6$。

首先可以发现，最高位相同的三个数异或和一定不为 0，异或和为 0 的情况只会在最高位不同时出现。

假设 f_i 表示最后一个数的最高位为 i 的合法序列数量，考虑转移。

A.Eminor Array

- 只选择最高位为 i 的数。有 $2^{2^i}-1$ 种方案。

A.Eminor Array

- 只选择最高位为 i 的数。有 $2^{2^i}-1$ 种方案。
- 上一个选择的最高位为 j。先考虑让 $(2^{2^i}-1)\cdot f_j \to f_i$，再减去不合法的情况。

A.Eminor Array

- 只选择最高位为 i 的数。有 $2^{2^i}-1$ 种方案。
- 上一个选择的最高位为 j。先考虑让 $(2^{2^i}-1)\cdot f_j \to f_i$，再减去不合法的情况。假设发生异或和为 0 的三个数分别为 x, y, z。由前述定理可知 x 最高位为 j，y, z 最高位为 i，且 $x < y < z$ 。

A.Eminor Array

- 只选择最高位为 i 的数。有 $2^{2^i}-1$ 种方案。
- 上一个选择的最高位为 j。先考虑让 $(2^{2^i}-1)\cdot f_j \to f_i$，再减去不合法的情况。
  假设发生异或和为 0 的三个数分别为 x,y,z。由前述定理可知 x 最高位为 j，y,z 最高位为 i，且 $x<y<z$ 。
  由于 $y=x\oplus z$，所以 $y<z$ 的充要条件为 z 的第 j 位为 1，故需要容斥减去的贡献为

$$\sum_{z=2^i}^{2^{i+1}-1}[j\in z]2^{(2^{i+1}-z-1)}\times f_j$$

  这个式子显然可以 $O(1)$ 计算。

故可以在 $O(n^2)$ 的时间复杂度内解决这个 DP。

A.Eminor Array

进一步简化式子，可以得到：

$$f_i = 2^{2^i} - 1 + \sum_{j=0}^{i-1} f_j \times (2^{2^i} - 1 - \frac{2^{2^i} - 1}{2^{2^j} + 1})$$

A.Eminor Array

进一步简化式子，可以得到：

$$f_i = 2^{2^i} - 1 + \sum_{j=0}^{i-1} f_j \times (2^{2^i} - 1 - \frac{2^{2^i} - 1}{2^{2^j} + 1})$$

将式子展开然后用前缀和优化即可，时间复杂度 $O(n \log MOD)$。

A.Eminor Array

进一步简化式子，可以得到：

$$f_i = 2^{2^i} - 1 + \sum_{j=0}^{i-1} f_j \times (2^{2^i} - 1 - \frac{2^{2^i} - 1}{2^{2^j} + 1})$$

将式子展开然后用前缀和优化即可，时间复杂度 $O(n \log MOD)$。

BONUS：如果上界不是 $2^n - 1$ 而是一个 $\leq 2^{10^6}$ 的任意一个数问题该如何解决？

K. String Divide II

简述题意

给定一个长度为 n 的字符串与参数 k。要求找到该字符串一个尽量长的子串，使得该子串时某字符串的 k 次循环。

数据范围：$n, k \leq 10^6$。

K. String Divide II

简述题意

给定一个长度为 n 的字符串与参数 k。要求找到该字符串一个尽量长的子串，使得该子串时某字符串的 k 次循环。

数据范围：$n, k \leq 10^6$。

考虑从大到小枚举字符串的长度 len，并寻找是否存在一个长度为 $len \times k$ 的子串满足条件。考虑将按长度 len 将字符串分割成 $\lceil \frac{n}{len} \rceil$ 段：

K. String Divide II

简述题意

给定一个长度为 n 的字符串与参数 k。要求找到该字符串一个尽量长的子串，使得该子串时某字符串的 k 次循环。

数据范围：$n, k \leq 10^6$。

考虑从大到小枚举字符串的长度 len，并寻找是否存在一个长度为 $len \times k$ 的子串满足条件。考虑将按长度 len 将字符串分割成 $\lceil \frac{n}{len} \rceil$ 段：

- 如果存在连续的 k 段相同，则 len 合法，答案为 $k \times len$。

K. String Divide II

简述题意

给定一个长度为 n 的字符串与参数 k。要求找到该字符串一个尽量长的子串，使得该子串时某字符串的 k 次循环。

数据范围：$n, k \leq 10^6$。

考虑从大到小枚举字符串的长度 len，并寻找是否存在一个长度为 $len \times k$ 的子串满足条件。考虑将按长度 len 将字符串分割成 $\lceil \frac{n}{len} \rceil$ 段：

- 如果存在连续的 k 段相同，则 len 合法，答案为 $k \times len$。
- 如果不存在连续的 $k-1$ 段相同，则 len 一定不合法。

K. String Divide II

所以主要的问题在于对一个连续的 $k-1$ 段相同的位置，我们能否对这个 $(k-1)\times len$ 的循环节进行左右扩展成为一个 $k\times len$ 循环节。

K. String Divide II

所以主要的问题在于对一个连续的 $k-1$ 段相同的位置，我们能否对这个 $(k-1)\times len$ 的循环节进行左右扩展成为一个 $k\times len$ 循环节。

而向左扩展的过程实际上是用一个前缀和作为一个子串的循环节求 LCS，这个问题可以利用 SA/SAM $O(1)$ 或 $O(\log n)$ 求解。向右扩展同理。

K. String Divide II

所以主要的问题在于对一个连续的 $k-1$ 段相同的位置，我们能否对这个 $(k-1)\times len$ 的循环节进行左右扩展成为一个 $k\times len$ 循环节。

而向左扩展的过程实际上是用一个前缀和作为一个子串的循环节求 LCS，这个问题可以利用 SA/SAM $O(1)$ 或 $O(\log n)$ 求解。向右扩展同理。

假设向左扩展了 $l1$ 的长度，向右扩展了 $l2$ 的长度，那么合法的充要条件即为 $l1+l2\geq k$。最后时间复杂度 $O(n\log n)$ 或 $O(n\log^2 n)$。

J. Lone Trail

题意简述

给定一个大小为 n 树以及参数 $a_1, a_2, \ldots, a_n$, $b_1, b_2, \ldots, b_n$。树上每个点初始点权为 b_i，每天点权会增加 a_i，有 m 次以下两种操作：

- 给定 x, u, v, w，在第 x 天，将 a_u 减去 w，将 a_v 加上 w。保证 u, v 之间有边，且操作后 a 非负。
- 给定 x。询问第 x 天的选择一点到其他点的权值乘上距离之和尽量小。即带权重心。

数据范围：$n, m \le 10^5$

J. Lone Trail

题意简述

给定一个大小为 n 树以及参数 $a_1, a_2, \ldots, a_n$, $b_1, b_2, \ldots, b_n$。树上每个点初始点权为 b_i，每天点权会增加 a_i，有 m 次以下两种操作：

- 给定 x, u, v, w，在第 x 天，将 a_u 减去 w，将 a_v 加上 w。保证 u, v 之间有边，且操作后 a 非负。
- 给定 x。询问第 x 天的选择一点到其他点的权值乘上距离之和尽量小。即带权重心。

数据范围：$n, m \le 10^5$

动态维护初始每个点的权值，用换根 DP 容易做到 $O(nm)$。

J. Long Trail

在没有修改的情况下，可以直接用换根 DP 维护每个点的花费关于时间的一次函数并维护凸包，查询在凸包上二分即可，时间复杂度 $O(n\log n)$

J. Long Trail

在没有修改的情况下，可以直接用换根 DP 维护每个点的花费关于时间的一次函数并维护凸包，查询在凸包上二分即可，时间复杂度 $O(n \log n)$

考虑每次修改会对这些一次函数造成什么影响，把所有修改转化为 x 时刻树上父亲向儿子移动 w ，不难发现每次会对一个子树进行相同修改 $(a_i = a_i - w, b_i = b_i + wx)$，对除该子树的其他点也进行相同的修改 $(a_i = a_i + w, b_i = b_i - wx)$

J. Long Trail

在没有修改的情况下，可以直接用换根 DP 维护每个点的花费关于时间的一次函数并维护凸包，查询在凸包上二分即可，时间复杂度 $O(n\log n)$

考虑每次修改会对这些一次函数造成什么影响，把所有修改转化为 x 时刻树上父亲向儿子移动 w ，不难发现每次会对一个子树进行相同修改 $(a_i=a_i-w, b_i=b_i+wx)$，对除该子树的其他点也进行相同的修改 $(a_i=a_i+w, b_i=b_i-wx)$

用差分把修改转化为只对子树的修改，在 dfs 序上表现为对一个区间的修改，此时可以对序列分块，对每个块维护凸包，整块打标记，散块暴力修改并重构凸包，查询二分每个块的凸包，可以做到 $O(n\sqrt{n}\log n)$

J. Long Trail

在没有修改的情况下，可以直接用换根 DP 维护每个点的花费关于时间的一次函数并维护凸包，查询在凸包上二分即可，时间复杂度 $O(n\log n)$

考虑每次修改会对这些一次函数造成什么影响，把所有修改转化为 x 时刻树上父亲向儿子移动 w ，不难发现每次会对一个子树进行相同修改 $(a_i = a_i - w, b_i = b_i + wx)$，对除该子树的其他点也进行相同的修改 $(a_i = a_i + w, b_i = b_i - wx)$

用差分把修改转化为只对子树的修改，在 dfs 序上表现为对一个区间的修改，此时可以对序列分块，对每个块维护凸包，整块打标记，散块暴力修改并重构凸包，查询二分每个块的凸包，可以做到 $O(n\sqrt{n}\log n)$

由于时间单调，对于没有重构过的块可以不用二分，至于重构的块其势能变化的和不会超过 $O(n\sqrt{n})$，每次重构后从头开始遍历即可，重构部分的排序用归并排序，复杂度降到 $O(n\sqrt{n})$