

Solution

作者 JDScript0117

注意事项

下文中可能一些很显然的结论也给了证明，如无需请自行跳过

T1

图论，数论，DP

Step 1.

发现对于每一个 i 向 p_i 建一条边，整体会形成很多环

$\times p$ 其实就是所有点沿有向边方向走 1 步

所以显然， p^k 就是所有点沿有向边方向走 k 步

容易发现， $f(p)$ 就是让你求出最小步数使所有点回到原来的位置

我们设 $check(p, x)$ 表示 p 所有点沿有向边方向走 x 步是否回到了原位

容易发现 $check(p, x) \rightarrow \forall x|y, check(p, y)$

同时我们假设图上环的长度为 $len_1, len_2, len_3, \dots, len_{cnt}$

那么容易发现， $check(p, x) = \forall 1 \leq i \leq cnt, len_i | x$

所以进一步意识到， $f(p) = lcm_{i=1}^{cnt} \{len_i\}$

Subtask 1.

我们设值域为 V ，根据上式发现了 $V \leq 60$

枚举全排列，对于每一个排列暴力运算，发现需要去重，可以用桶，最终时间复杂度 $\Theta(n! \times n + V)$ ，剪剪枝卡卡常可做

Step 2.

发现需要去重，只需要对于每一个 x 去判断是否 $\exists p \in P_n, f(p) = x$

会发现环的长度和等于 n ，所以可以等价于

$$\exists len_1, len_2, len_3, \dots, len_{cnt} \geq 1, \sum_{i=1}^{cnt} len_i = n, lcm_{i=1}^{cnt} \{len_i\} = x$$

将后面两个条件交换一下顺序，就变成了

$$\exists len_1, len_2, len_3, \dots, len_{cnt} \geq 1, lcm_{i=1}^{cnt} \{len_i\} = x, \sum_{i=1}^{cnt} len_i = n$$

会发现 $len = 1$ 时对 lcm 做不出贡献，只对和有贡献，所以可以转化为如下形式

$$\exists len_1, len_2, len_3, \dots, len_{cnt} \geq 2, lcm_{i=1}^{cnt} \{len_i\} = x, \sum_{i=1}^{cnt} len_i \leq n$$

我们令 $g(x) = \min_{len_1, len_2, len_3, \dots, len_{cnt} \geq 2, lcm\{len_i\}_{i=1}^{cnt} = x} \sum_{i=1}^{cnt} len_i$

那么其实上面的条件就可以转化为 $g(x) \leq n$, 现在的问题就变成了快速求 $g(x)$

发现 $\exists 1 \leq x < y \leq cnt, len_x \nmid len_y$ 时一定不是最小, 证明如下

$$\begin{aligned} lcm\{len_i\}_{i=1}^{cnt} &= lcm(lcm\{lcm\{len_i\}_{i=1}^{x-1}, lcm\{len_i\}_{i=x+1}^{y-1}, lcm\{len_i\}_{i=y+1}^{cnt}\}, lcm(len_x, len_y)) \\ lcm(len_x, len_y) &= lcm(len_x, \frac{len_y}{gcd(len_x, len_y)}) \\ \text{令 } len_y &= \frac{len_y}{gcd(len_x, len_y)}, lcm\{len_i\}_{i=1}^{cnt} \text{ 不变} \\ gcd(len_x, len_y) &> 1 \\ &\rightarrow \sum_{i=1}^{y-1} len_i + \sum_{i=y+1}^{cnt} len_i + len_y > \sum_{i=1}^{y-1} len_i + \sum_{i=y+1}^{cnt} len_i + \frac{len_y}{gcd(len_x, len_y)} \end{aligned}$$

然后我们就可以确定 $\forall 1 \leq x < y \leq cnt, len_x \perp len_y$

然后我们又可以知道 $lcm\{len_i\}_{i=1}^{cnt} = \prod_{i=1}^{cnt} len_i$

发现 $\exists 1 \leq pos \leq cnt, \exists x \perp y \geq 2, len_{pos} = x \times y$ 是一定不是最小, 证明如下

$$\begin{aligned} x \perp y &\rightarrow x \neq y \\ \text{不妨设 } x < y, &\text{ 可得 } len_{pos} = x \times y \geq 2 \times y = y + y > x + y \\ len_{pos} = x \times y, &\forall 1 \leq z \leq cnt \wedge z \neq pos, len_{pos} \perp len_z \\ &\rightarrow \forall 1 \leq z \leq cnt \wedge z \neq pos, x, y \perp len_z \\ \text{令 } len_{pos} = x, len_{cnt+1} = y, &cnt = cnt + 1, \prod_{i=1}^{cnt} len_i \text{ 不变}, \sum_{i=1}^{cnt} len_i \text{ 变小} \end{aligned}$$

于是乎我们发现 $\forall 1 \leq pos \leq cnt, \forall x \perp y \geq 2, len_{pos} \neq x \times y$

我们只取后面半截, 研究 $\forall x \perp y \geq 2, z \neq x \times y$ 发现等效于 $\exists p \in \mathbb{P}, k \in \mathbb{N}, p^k = z$, 其中 \mathbb{P} 表示素数集, 证明如下

证明两条件等效, 只需证明充分必要性即可

充分性证明: 证明 $\exists p \in \mathbb{P}, k \in \mathbb{N}, p^k = z \rightarrow \forall x \perp y \geq 2, z \neq x \times y$

$$\begin{aligned} \exists p \in \mathbb{P}, k \in \mathbb{N}, p^k = z &\rightarrow \forall x \geq 2, x|z, p|x \\ &\rightarrow \forall x, y \geq 2, x, y|z, x \nmid y \\ &\rightarrow \forall x, y \geq 2, x \times y = z, x \nmid y \\ &\leftrightarrow \forall x \perp y \geq 2, z \neq x \times y \end{aligned}$$

必要性证明: 考虑反证法, 即证 $\forall p \in \mathbb{P}, k \in \mathbb{N}, p^k \neq z \rightarrow \exists x \perp y \geq 2, z = x \times y$

$\forall p \in \mathbb{P}, k \in \mathbb{N}, p^k \neq z$

$$\begin{aligned} &\rightarrow \exists cnt \geq 2, p_1 \neq p_2 \neq p_3 \neq \dots \neq p_{cnt} \in \mathbb{P}, k_1, k_2, \dots, k_{cnt} \in \mathbb{Z}^*, \prod_{i=1}^{cnt} p_i^{k_i} = z \\ &\rightarrow p_1^{k_1} \perp \prod_{i=2}^{cnt} p_i^{k_i} \geq 2, z = p_1^{k_1} \times \prod_{i=2}^{cnt} p_i^{k_i} \end{aligned}$$

根据以上结论, 可以证明当 $x = p_1^{\alpha_1} \times p_2^{\alpha_2} \times p_3^{\alpha_3} \times \dots \times p_{cnt}^{\alpha_{cnt}} (\forall 1 \leq i \leq cnt, p_i \in \mathbb{P}, \alpha_i \in \mathbb{Z}^*)$

时, $g(x) = \sum_{i=1}^{cnt} p_i^{\alpha_i}$

Subtask 2A.

发现其实对于每一个 x ，我们只需要分解质因数即可，发现 $V \leq 360360$ ，时间复杂度 $\Theta(V \times \ln V)$

Subtask 2B.

发现最初使用线性筛，可以在筛时转移 $a(x)$ 表示 x 的 $p_{cnt}^{\alpha_{cnt}}$ ，发现 $g(x) = g(\frac{x}{a(x)}) + a(x)$ ，线性转移即可，时间复杂度 $\Theta(V)$

Subtask 3.

看到了线性转移，考虑使用 dp ，我们将 n 以内的质数排序

$dp_{i,j}$ 表示只使用 $p_1, p_2, p_3, \dots, p_i$ 这些质数， g 等于 j 的数之和

写出转移式 $dp_{i,j} = dp_{i-1,j} + p_i \times dp_{i-1,j-p_i} + p_i^2 \times dp_{i-1,j-p_i^2} + p_i^3 \times dp_{i-1,j-p_i^3} + \dots$

我们记 $m(n) = \sum_{p \in \mathbb{P}} \lfloor \log_p n \rfloor$ ，时间复杂度应为 $\Theta(n \times m(n))$ 可以算出 $m(10^5) = 9700$ ，但常数较小

滚动优化后，空间复杂度为 $\Theta(n)$

T2

分块，图论

Subtask 1.

暴力判断任意两个询问，时间复杂度 $\Theta(q^2)$

Subtask 2.

暴力判断或覆盖花环内所有点，时间复杂度 $\Theta(q \times r^2)$

Subtask 3.

发现我们可以对于每一列用 `set` 存下当前留下来的花环

发现判断时只需判断每一列纵坐标比它小的最大的和比它大的最小的，时间复杂度 $\Theta(q \times n \times \ln m)$

Subtask 4.

会发现我们在判断 i 时，其实是枚举 j 判断的 $|x_i - x_j| + |y_i - y_j|$ 与 $r_i + r_j$ 的关系

每一个都判一遍，必定繁琐考虑移项，变成 $\min_j \{|x_i - x_j| + |y_i - y_j| - r_j\}$ 与 r_i 的关系

发现 $|x_i - x_j| + |y_i - y_j|$ 可以看做将相邻点之间建边权为 1 的无向边，以 (x_j, y_j) 为源点跑单源最短路的结果

拓展会发现， $\min_j \{|x_i - x_j| + |y_i - y_j| - r_j\}$ 和我们再建一个超级源点，向每一个已经保留的花环 (x_j, y_j) 连一条权值为 $-r_j$ 的有向边后对于 (x_i, y_i) 的单源最短路

发现被覆盖的点只能可能被这个花环转移，所以在保留这个花环之后就直接赋值了，会发现边界会出现很多距离为 -1 的点

发现边权永远为 1，每一次只需要使用 BFS，初始点集合就记为所有边界点，单源最短路时间复杂度为 $\Theta(n \times m)$ ，同时还发现每个边界点在上次最短路跑过之后就不用再跑了，还可以大大优化

考虑将询问分块，设一个阈值 V ，每 V 次询问后跑一遍单源最短路，所以我们每个询问只需要判断最短路，再判断上次跑最短路之后的花环与这个花环的影响，时间复杂度 $\Theta(\frac{q}{V} \times n \times m + q \times V)$ ，标程的 V 设的 1000

T3

数学，数据结构

Step 1.

根据裴蜀定理

将

$$\forall x \in \mathbb{Z}, \exists k_l, k_{l+1}, k_{l+2}, \dots, k_r \in \mathbb{Z}, k_l \times F_l + k_{l+1} \times F_{l+1} + k_{l+2} \times F_{l+2} + \dots + k_r \times F_r = x$$

$$\text{转化为 } \gcd_{i=l}^r \{F_i\} = 1$$

Subtask 1, 2, 3.

根据特定 a, b, c 推出 f 的表达式，暴力计算答案，时间复杂度约为

$$\Theta\left(\left(\sum_{i=1}^n cnt_i\right) \times \ln d + n^2 \times \ln m\right)$$

Subtask 4, 5, 6.

继承上一组的想法，计算答案时使用 ST 表+二分，时间复杂度应为

$$Theta\left(\left(\sum_{i=1}^n cnt_i\right) \times \ln d + q \times \ln d + q \times n \times \ln n \times \ln m\right)$$

Step 2.

容斥，发现我们只需要统计 $\gcd_{i=l}^r \{F_i\} > 1$ 的区间数量

Subtask 7.

根据 $m = 2$ ，发现只在意 F 的奇偶性，容易维护 F ，留给一些 bitset 做法

统计答案我们只在意 2 的连续段，修改时只会合并或分裂，留给一些 set 做法

Step 3.

思考如何带修改快速统计答案

首先想到将区间 $[l, r]$ 看做点对 (l, r) 想到 分治

如果我们要统计 $l, r \in [l_0, r_0]$ 的点对，可以先统计 $l, r \in [l_0, mid]$ 与 $l, r \in [mid + 1, r_0]$ 内的所有点对，再统计 $l \in [l_0, mid], r \in [mid, r_0]$ 的点对

我们只在意如何统计 $l \in [l_0, mid], r \in [mid, r_0]$ 的点对，其它的类似递归

$$\text{容易发现此时 } \gcd_{i=l}^r \{F_i\} > 1 \text{ 变为了 } \gcd_{i=l}^{mid} \{F_i\} \nmid \gcd_{i=mid+1}^r \{F_i\}$$

会发现, 对于一段区间 $[l, r]$, 所有区间内的 x 中, $\gcd_{i=l}^x\{F_i\}, \gcd_{i=x}^r\{F_i\}$ 只有 $\Theta(\ln m)$ 种值

于是发现我们可以开 线段树, 每个节点维护 ans 表示这个节点的区间内点对的数量, $gcdsum$ 表示这个节点的区间内的最大公约数

每个节点还维护 $lft[20][2], rgt[20][2]$ 表示从左往右, 从右往左不同的 gcd 值, 与达成这个值的终止点
 $pushUp()$ 时 $lft, rgt, gcdsum$ 容易合并, ans 应该是两儿子 ans 之和再加上中间的点对, 发现单调性之后中间的点对考虑用 双指针 去统计贡献, 最终时间复杂度是 $\Theta(\ln^2 m)$

由于是线段树, 所以单次修改时间复杂度应为 $\Theta(\ln n \times \ln^2 m)$

Subtask 8, 9, 10.

还是一样的推式子, 优化统计答案, 时间复杂度

$$\Theta\left(\left(\sum_{i=1}^n cnt_i\right) \times \ln d + q \times \ln d + q \times \ln n \times \ln^2 m\right)$$

Subtask 11, 12.

发现输入时得写一颗区间乘区间和的线段树, 统计答案分别对应的 Subtask 1, 2, 3 与 Subtask 4, 5, 6 的时间复杂度

Step 4.

关注一下 $a = 1, b = 1, c = 0$, 不知道有没有人做过这样一道题

给定一个整数 n , 要求在 $\Theta(\ln n)$ 的时间复杂度下求出 $FBNQ(n)$

一道很典的 矩阵快速幂

我们构造一个长度为二的数组 $[FBNQ(i), FBNQ(i-1)]$ 发现它要想向后转移成 $[FBNQ(i+1), FBNQ(i)]$ 其实也就是 $[FBNQ(i) + FBNQ(i-1), FBNQ(i)]$, 这是一个一次关系映射, 所以可以构造出矩阵

$$\begin{bmatrix} FBNQ(i) & FBNQ(i-1) \\ 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} FBNQ(i) + FBNQ(i-1) & FBNQ(i) \\ 0 & 0 \end{bmatrix}$$

于是我们就可以根据矩阵的乘法结合律来实现了

Subtask 13, 16.

优化计算 $FBNQ$ 的过程, 统计答案分别对应 Subtask 4, 5, 6 与 Subtask 8, 9, 10 的时间复杂度

Step 5.

发现 $c = 0$, 考虑将矩阵从特殊到一般一下

还是构造一个长度为二的数组 $[f(i), f(i-1)]$ 发现它要向后转移成 $[a \times f(i) + b \times f(i-1), f(i)]$, 构造出如下矩阵

$$\begin{bmatrix} f(i) & f(i-1) \\ 0 & 0 \end{bmatrix} \times \begin{bmatrix} a & 1 \\ b & 0 \end{bmatrix} = \begin{bmatrix} a \times f(i) + b \times f(i-1) & f(i) \\ 0 & 0 \end{bmatrix}$$

Subtask 14, 17.

优化计算 f 的过程, 统计答案分别对应 Subtask 4, 5, 6 与 Subtask 8, 9, 10 的时间复杂度

Step 6.

考虑继续扩展矩阵

构造一个长度为三的数组 $[f(i), f(i-1), 1]$ 发现它要向后转移成 $[a \times f(i) + b \times f(i-1) + c, f(i), 1]$ ，构造出如下矩阵

$$\begin{bmatrix} f(i) & f(i-1) & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} a & 1 & 0 \\ b & 0 & 0 \\ c & 0 & 1 \end{bmatrix} = \begin{bmatrix} a \times f(i) + b \times f(i-1) + c & f(i) & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Subtask 15, 18.

优化计算 f 的过程，统计答案分别对应 Subtask 4, 5, 6 与 Subtask 8, 9, 10 的时间复杂度

Step 7.

不知道你们有没有做过这样一道题

给定一个长度为 n 的数组 a ，有 q 次询问，要么将 $[l, r]$ 内增加 x ，要么查询 $\sum_{i=l}^r FBNQ(a_i)$

这就相当于上面给的那道题的加强版了，做法是考虑矩阵乘法具有分配律

开一棵 线段树，每个叶子节点维护区间内的矩阵和，于是就变成了区间乘区间和的线段树了

注意要加 *lazy_tag*，矩阵的 1 一般是指形如下面这种矩阵

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \dots$$

Subtask 19, 20, 21.

回归这道题，发现要维护 F 可以像 Step 7 中的思路一样开一棵 主席树

矩阵乘法构造分别对应 Step 4，Step 5 与 Step 6 的方案

同时统计答案采用 Subtask 8, 9, 10 中的方案，假设矩阵的大小为 $size \times size$

最终的时间复杂度应为 $\Theta\left(\left(q + \sum_{i=1}^n cnt_i\right) \times size^3 \times (\ln k + \ln d) + q \times \ln n \times \ln^2 m\right)$

T4

数据结构，图论

Subtask 1.

发现限制要求就是任意两个 T 有交集的方案不能同时不选

考虑将所有 T 有交集的方案对之间建无向边，就变成了每条边两端不能同时不选

用 邻接表，暴力，卡卡能过，时间复杂度约为 $\Theta(2^n \times n^2)$

Subtask 2.

发现所有方案中只能有一种方案不选

那我们删的方案一定删之前最小异或的那一对点其中之一

考虑如何求删之前最小异或的那一对点

提取到关键词最小异或, 01Trie 我不再再说了吧

找到了之后可以考虑枚举删哪个点跑一遍

时间复杂度 $\Theta(n \times \ln n)$

Subtask 3.

先像 Subtask1 一样建边, 假设我们建了 m 条

看到了每条边两端不能同时不选, 有 2-SAT 的感觉

发现决策有单调性, 考虑二分答案, 值域我们记为 A

对于二分到的答案 x , 我们考虑用 2-SAT 做

假设 u, v 之间有一条边, 则建 $(u, 0) \rightarrow (v, 1)$ 和 $(u, 1) \rightarrow (v, 0)$

假设 $t_u \oplus t_v < x$, 则建 $(u, 1) \rightarrow (v, 0)$ 和 $(u, 0) \rightarrow (v, 1)$

假设点数为 V , 边数为 E , 则 Tarjan 的时间复杂度为 $\Theta(V + E)$

发现 m 是 $\Theta(n^2)$ 级别的, V 是 $\Theta(n)$ 级别的, E 是 $\Theta(n^2 + m)$ 级别的

于是最终时间复杂度应为 $\Theta(n^2 \times \ln A)$

Subtask 4.

发现 m 是 $\Theta\left(\sum_{i=1}^n \ln n_i\right)$ 级别的, V 是 $\Theta(n)$ 级别的, E 是 $\Theta(n^2 + m)$ 级别的

所以我们主要想把 E 优化一下

我们其实要优化的就是 $t_u \oplus t_v < x$ 的部分

考虑建 01Trie 发现对于 t_u, t_v 应该在 $\ln A$ 个节点的子树内

那么我们对于这个 01Trie, 假设 v 是 u 的儿子, 则建 $(u, 0) \rightarrow (v, 0)$

然后对于 t_u , 我们将这 $\ln A$ 个节点找出来, 假设我们找出的这些节点的其中之一是 v , 则建 $(u, 1) \rightarrow (v, 0)$

可以证明, 这样建图是等效的, 优化建图后 E 变为了 $\Theta(n \times \ln A + m)$ 级别的

最终时间复杂度为 $\Theta\left(n \times \ln^2 A + \left(\sum_{i=1}^n \ln n_i\right) \times \ln A\right)$

Subtask 5.

发现 m 是 $\Theta\left(\left(\sum_{i=1}^n \ln n_i\right)^2\right)$ 级别的, V 是 $\Theta(n)$ 级别的, E 是 $\Theta(n \times \ln A + m)$ 级别的

所以我们主要想把 m 优化一下

发现对于不同的 T , 我们将包含它的方案集记作 F_T , 发现我们对 $u \in F_T$, 我们要将所有 $v \neq u \in F_T$ 建边

考虑将 F_T 排序，然后每一个点连的边就是两个区间，建一棵 线段树，可以发现，每个区间就是最多 $\Theta(\ln |F_T|)$ 级别个点的子树内

思路差不多，对于线段树上每一个节点，假设 u 是 v 的父亲，则建 $(u, 1) \rightarrow (v, 1)$

然后对于每一个点 u ，我们将这 $\Theta(\ln |F_T|)$ 个节点找出来，假设我们找出的这些节点的其中之一是 v ，则建 $(u, 0) \rightarrow (v, 1)$

这样 m 是 $\Theta\left(\sum_T |F_T| \times \ln |F_T|\right)$ 级别的，化简后即是 $\Theta\left(\left(\sum_{i=1}^n \ln n_i\right) \times \ln\left(\sum_{i=1}^n \ln n_i\right)\right)$ 级别的了

最终时间复杂度为 $\Theta\left(n \times \ln^2 A + \left(\sum_{i=1}^n \ln n_i\right) \times \ln\left(\sum_{i=1}^n \ln n_i\right) \times \ln A\right)$ 了