

# 愚蠢的在线法官 解题报告

宁波市镇海中学 潘佳奇

October 3, 2021

## Contents

|          |                  |          |
|----------|------------------|----------|
| <b>1</b> | <b>题目描述</b>      | <b>2</b> |
| 1.1      | 简要题意 . . . . .   | 2        |
| 1.2      | 数据范围 . . . . .   | 2        |
| <b>2</b> | <b>题解</b>        | <b>2</b> |
| 2.1      | 算法一 . . . . .    | 2        |
| 2.2      | 算法二 . . . . .    | 3        |
| 2.3      | 算法三 . . . . .    | 3        |
| 2.4      | 算法四 . . . . .    | 3        |
| 2.5      | 算法五 . . . . .    | 5        |
| <b>3</b> | <b>总结</b>        | <b>9</b> |
| 3.1      | 思想总结 . . . . .   | 9        |
| 3.2      | 出题过程 . . . . .   | 9        |
| 3.3      | 对题目的评价 . . . . . | 10       |

# 1 题目描述

## 1.1 简要题意

给定一棵  $n$  个点的有根树和每个点的点权  $v_i$ ，令根为节点 1。给定一个长度为  $k$  的数组  $A$ ，构造阶为  $k$  的方阵  $B$  其中  $b_{i,j} = v_{\text{LCA}(A_i, A_j)}$ ，即：

$$B = \begin{bmatrix} v_{\text{LCA}(A_1, A_1)} & v_{\text{LCA}(A_1, A_2)} & \cdots & v_{\text{LCA}(A_1, A_k)} \\ v_{\text{LCA}(A_2, A_1)} & v_{\text{LCA}(A_2, A_2)} & \cdots & v_{\text{LCA}(A_2, A_k)} \\ \vdots & \vdots & \ddots & \vdots \\ v_{\text{LCA}(A_k, A_1)} & v_{\text{LCA}(A_k, A_2)} & \cdots & v_{\text{LCA}(A_k, A_k)} \end{bmatrix}$$

其中  $\text{LCA}(x, y)$  为节点  $x$  和  $y$  的最近公共祖先。求这个方阵的行列式，答案对 998244353 取模。

## 1.2 数据范围

对于所有数据，满足  $1 \leq n, k \leq 5 \times 10^5$ ， $v_i \in [0, 998244353)$ ， $A_i \in [1, n]$ 。

子任务 1（3 分），保证  $k > n$ 。

子任务 2（6 分），保证  $n \leq 10$ 。

子任务 3（11 分），保证  $k \leq 600$ 。

子任务 4（29 分），保证  $n \leq 3000$ 。

子任务 5（16 分），保证  $A$  是  $1 \dots n$  的排列。

子任务 6（35 分），没有特殊条件。

# 2 题解

## 2.1 算法一

由鸽巢原理，当  $k > n$  的时候，行列式中必有两行相同，此时行列式值为 0。可以通过子任务 1，期望得分 3。

## 2.2 算法二

根据行列式的定义，直接求行列式的值，使用  $O(n+Q)$  的 LCA 求法，时间复杂度  $O(k!+n)$ 。

结合算法一，可以通过子任务 1,2，期望得分 9。

## 2.3 算法三

使用高斯消元求行列式值，使用  $O(n+Q)$  的 LCA 求法，时间复杂度  $O(k^3+n)$ 。结合算法一，可以通过子任务 1,2,3，期望得分 20。

## 2.4 算法四

### 2.4.1 从 gcd 矩阵说起

先来介绍另一个问题：

计算

$$A = \begin{bmatrix} \gcd(1,1) & \gcd(1,2) & \cdots & \gcd(1,n) \\ \gcd(2,1) & \gcd(2,2) & \cdots & \gcd(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ \gcd(n,1) & \gcd(n,2) & \cdots & \gcd(n,n) \end{bmatrix}$$

的行列式。

其答案是  $\varphi(1)\varphi(2)\dots\varphi(n)$ 。

考虑到  $\gcd(a,b)$  的意义，是最大的  $t$  使得  $t|a$  且  $t|b$ 。

对于“且”即  $\wedge$  的条件，可以使用乘法来叙述： $[t|a \wedge t|b]$  即  $[t|a][t|b]$ ，而满足次条件的所有  $t$  都满足  $t|\gcd(a,b)$ 。因为  $n = \sum_{i=1}^n [i|n]\varphi(i)$ ，那么  $\gcd(a,b) = \sum_{i=1}^n [i|a][i|b]\varphi(i)$ 。

因此  $A$  可以表示为两个矩阵的积：

$$A = BC$$

$$b_{i,j} = \begin{cases} \varphi(j) & j|i \\ 0 & j \nmid i \end{cases}$$

$$c_{i,j} = \begin{cases} 1 & i|j \\ 0 & i \nmid j \end{cases}$$

根据方阵乘法的公式  $|A| = |B||C|$ ，只需要分别计算  $|B|$  和  $|C|$ 。

而  $B, C$  是三角矩阵，因此  $|B| = \varphi(1)\varphi(2)\dots\varphi(n)$ ， $|C| = 1$ 。

所以  $|A| = \varphi(1)\varphi(2)\dots\varphi(n)$ 。

### 2.4.2 回到 LCA

考虑到 LCA 的条件同样可以表示成两个条件的乘积。令  $a \rightarrow b$  表示  $a$  是  $b$  的祖先， $a \nrightarrow b$  就表示  $a$  不是  $b$  的祖先。那么令  $v_{\text{LCA}(u,v)} = \sum_{i=1}^n f_i[i \rightarrow a][i \rightarrow b]$ ，需要解  $\sum_{i=1}^n [i \rightarrow u]f_i = v_u$ ，记  $p_u$  为  $u$  的父亲，解得  $f_u = v_u - v_{p_u}$ 。

先考虑题目给出的序列  $A$  是  $1 \sim n$  的排列时的解法：可以将  $A$  排序，因为先交换行列式的一行，再交换行列式的一列，行列式值不变。这保证了将  $A$  排序行列式值不变。

此时假设  $A = [1, 2, \dots, n]$ ，那么类似 gcd 矩阵，题目中的  $B$  可以表示为两个矩阵  $C, D$  的乘积：

$$B = CD$$

$$c_{i,j} = \begin{cases} v_j - v_{p_j} & j \rightarrow i \\ 0 & j \nrightarrow i \end{cases}$$

$$d_{i,j} = \begin{cases} 1 & i \rightarrow j \\ 0 & i \nrightarrow j \end{cases}$$

特殊地，记根  $r$  的  $p_r = 0$ ，记  $v_0 = 0$ ，此时由于  $C, D$  是三角矩阵， $|C| = \prod_{i=1}^n (v_i - v_{p_i})$ ， $|D| = 1$ ，因此可以  $O(n)$  计算出  $|B|$ 。

结合算法一、三，期望得分 36。

## 2.5 算法五

### 2.5.1 扩展到 $k < n$

当  $k < n$  时  $A$  不能表示成两个方阵的积了：在  $A$  所有节点组成的虚树中，有可能存在点  $u \notin A$ 。此时即便将不在  $A$  中的点去掉，也无法正确计算答案。

但是，类似地可以使用  $k \times n$  的矩阵和  $n \times k$  的矩阵相乘：

$$B_{k \times k} = C_{k \times n} D_{n \times k}$$

$$c_{i,j} = \begin{cases} v_j - v_{p_j} & j \rightarrow A_i \\ 0 & j \nrightarrow A_i \end{cases}$$

$$d_{i,j} = \begin{cases} 1 & i \rightarrow A_j \\ 0 & i \nrightarrow A_j \end{cases}$$

此时计算行列式值不能使用之前的方法，因为  $C, D$  不是方阵。

尝试使用 Cauchy-Binet 公式<sup>1</sup> 来计算大小为  $n \times m$  和  $m \times n$  矩阵相乘的行列式值：

$$|AB| = \begin{cases} 0 & n > m \\ |A||B| & n = m \\ \sum_{S \subseteq \{1, \dots, m\}, |S|=n} |A_{\{1, \dots, n\}, S}| |B_{S, \{1, \dots, n\}}| & n < m \end{cases}$$

记在  $A$  中的点为特殊点，那么其组合意义为：选定一个大小为  $k$  的集合，称这个集合为未匹配点的集合。再用特殊点与集合中的点匹配，同时要求每个特殊点匹配的点是其祖先。每个匹配方案权值为被匹配点的  $v_u - v_{p_u}$  的乘积，乘上逆序数的贡献。将贡献累加可以得到行列式的值。

但是，即便按照这样的方法理解，由于逆序数的存在，子式还是难以计算。

### 2.5.2 考虑正负抵消

很多情况下，计算图相关行列式的时候，会存在一些方案由于符号不同而互相抵消：当方案中存在两个特殊点  $u, v$ ，分别匹配  $u', v'$ ，那么有  $u' \rightarrow u, v' \rightarrow v$ 。

<sup>1</sup>[https://en.wikipedia.org/wiki/Cauchy-Binet\\_formula](https://en.wikipedia.org/wiki/Cauchy-Binet_formula)

当  $u' \rightarrow v, v' \rightarrow u$  时, 交换  $u', v'$ , 也是一个合法的匹配方案。此时逆序数奇偶性发生了变化, 但是被匹配点的集合没有发生变化。那么猜想: 对于一个方案, 只要存在两个点  $u, v$  满足其匹配点可以互相交换, 这个方案就可以和另一个方案进行抵消。为了证实这个猜想, 我们要指定一个交换方案, 即构造一个函数  $f$ , 使得交换后的结果为  $f(X) = Y$ 。

对于所有存在可交换匹配点的匹配方案  $X$ , 找到其字典序最小的二元组  $(u, v)$  ( $u \neq v$ ) 使得  $u$  和  $v$  的匹配点可以交换。其中二元组的字典序比较以  $dfn_{X_u}$  (点  $u$  的匹配点  $X_u$  的 dfs 序) 为标准。

找到二元组  $(u, v)$  后, 记匹配方案  $X$  交换  $u, v$  对应的匹配点后得到的匹配方案为  $f(X)$ 。为了证明猜想, 首先要证明对于任意两个匹配  $X_1, X_2$  满足  $X_1 \neq X_2 \Rightarrow f(X_1) \neq f(X_2)$ :

*Proof.* 假设  $f(X_1) = f(X_2)$ 。

记  $X_1$  交换的是  $(u_1, v_1)$ ,  $X_2$  是  $(u_2, v_2)$ 。

当  $(u_1, v_1) = (u_2, v_2)$ , 则与  $X_1 \neq X_2$  矛盾。

如果  $u_1, v_1, u_2, v_2$  是两两不同的, 不妨假设  $(u_1, v_1) < (u_2, v_2)$ , 那么  $(u_1, v_1)$  是  $X_2$  能交换的, 与字典序最小矛盾。

如果  $u_1, v_1, u_2, v_2$  刚好有一对相同, 那么存在  $u, v, w, a, b, c$  使得

1.  $u \rightarrow a, u \rightarrow b$
2.  $v \rightarrow a, v \rightarrow b, v \rightarrow c$
3.  $w \rightarrow b, w \rightarrow c$

其中  $a, b, c$  是关键点,  $u, v, w$  是树上的点,  $b$  是重复的点。在  $f(X_1)$  中  $a$  匹配  $u$ ,  $b$  匹配  $v$ ,  $c$  匹配  $w$ 。  $a \in \{u_1, v_1\}, c \in \{u_2, v_2\}$ 。

由于  $u, w$  都是  $b$  的祖先, 所以总有一个点是  $\text{LCA}(a, b), \text{LCA}(c, b)$  两个点的祖先, 所以  $u \rightarrow c, w \rightarrow a$  两者至少一者成立。两者如果同时成立, 那么  $(u_1, v_1)$  是  $X_2$  能交换的, 与字典序最小矛盾。所以此时假设两者恰好一者成立。

由字典序比较方式, 得  $dfn_u < dfn_w$ , 即  $u \rightarrow c$  成立。

在  $X_2$  中匹配关系为  $u \rightarrow a, w \rightarrow b, v \rightarrow c$ 。因为  $X_2$  交换的是  $\min\{(w, v), (v, w)\}$ , 字典序一定比  $\min\{(u, v), (v, u)\}$  大, 而  $\min\{(u, v), (v, u)\}$  是  $X_2$  能交换的, 与字典序最小矛盾。

所以对于任意  $X_1, X_2$ , 有  $f(X_1) \neq f(X_2)$ 。

□

对于任意逆序数为奇数的匹配方案都能对应到唯一的逆序数为偶数的匹配方案, 偶数的同理, 所以  $f$  满足方案到方案的双射。

所有存在可交换匹配点的方案中, 记逆序数为奇数的方案集合为  $O$ , 偶数的为  $E$ ,  $g(x)$  为匹配方案的权值。那么有

$$\begin{aligned}
 & \sum_{x \in O} g(x) + \sum_{x \in E} g(x) \\
 &= - \sum_{x \in O} g(f(x)) + \sum_{x \in E} g(x) \\
 &= \sum_{x \in E} g(x) - \sum_{x \in E} g(x) \\
 &= 0
 \end{aligned}$$

也就是说, 对于所有存在可交换匹配点的方案的贡献和是 0。

### 2.5.3 考虑剩下的方案匹配的唯一性

如果所有方案中都存在点无法匹配, 那么贡献为 0。

称不存在可交换匹配点且存在匹配的方案为有贡献的方案。

此时断言, 对于有贡献的方案, 每个点只能选择唯一的祖先, 即匹配是唯一的:

*Proof.* 匹配的唯一性

此时对于任意有贡献的方案, 一定能取出一个匹配  $f$ , 假设存在匹配  $g \neq f$ , 此时存在序列  $S, T$ , 满足  $|S| = |T| = m$ , 且  $S, T$  都是没有重复元素的序列, 使得对于  $1 \leq i \leq m$ , 满足:

1.  $T_i \rightarrow S_i$
2.  $T_{(i \bmod m)+1} \rightarrow S_i$

由于有

1.  $T_i \rightarrow S_i$

$$2. T_{(i \bmod m)+1} \rightarrow S_i$$

$$3. T_{(i \bmod m)+1} \rightarrow S_{(i \bmod m)+1}$$

如果有  $T_i \rightarrow S_{(i \bmod m)+1}$ ，则存在可交换匹配点，与有贡献的条件矛盾。所以  $T_i \nrightarrow S_{(i \bmod m)+1}$ 。

由于有

$$1. T_i \rightarrow S_i$$

$$2. T_i \nrightarrow S_{(i \bmod m)+1}$$

$$3. T_{(i \bmod m)+1} \rightarrow S_i$$

$$4. T_{(i \bmod m)+1} \rightarrow S_{(i \bmod m)+1}$$

当  $T_i \rightarrow T_{(i \bmod m)+1}$ ，就会有  $T_i \rightarrow S_{(i \bmod m)+1}$ ，矛盾。

当  $T_{(i \bmod m)+1} \nrightarrow T_i$ ，就会有  $T_{(i \bmod m)+1} \nrightarrow S_i$ ，矛盾。

所以  $T_{(i \bmod m)+1} \rightarrow T_i$ ，由于  $T_i$  互不相同，所以树上的深度  $dep_{T_{(i \bmod m)+1}} < dep_{T_i}$ 。

由传递性得到  $dep_{T_i} < dep_{T_i}$ ，矛盾。

所以匹配是唯一的。 □

### 2.5.4 转化为树形 DP

因为每个点只能匹配其祖先，对于大小为  $k$  的有唯一匹配的集合的选定，可以在树上做 DP。记  $f_{i,j}$  为在子树  $i$  中，还有  $j$  个关键点未匹配的权值和，转移的正确性由乘法分配律保证。

这道题 DP 的方法为将子树的信息合并，并添加一个新的点作为这棵新的树的根，从根向子树连边。那么在合并子树的时候，可以将  $f_{i,j}$  的  $j$  相加。在加入一个树上的点的时候，可以决策这个点是否与未匹配的关键点匹配。

这里断言，在子树合并后， $j$  最多为 1。因为在合并子树之后，如果未匹配的关键点  $> 1$ ，那么由于所有未匹配的关键点都会在祖先匹配完，那么一定会出现一对关键点，其可以交换匹配点。

那么有两种转移：



1. 子树的合并（背包模型）： $f'_n = \sum_{j+k=n} f_j \times g_k$

2. 树上点  $u$  的匹配

- 当这个点是关键点时，子树必须匹配完，此时  $f'_1 = f_0, f'_0 = f_0(v_u - v_{p_u})$ 。
- 当这个点不是关键点的时候， $f'_1 = f_1, f'_0 = f_0 + f_1(v_u - v_{p_u})$ 。

### 2.5.5 考虑逆序数的影响

回到在计算子式的时候，逆序数的问题。因为  $|A_{\{1,\dots,k\},S}|, |B_{S,\{1,\dots,k\}}|$  的匹配选择是相同的，也就是其逆序数奇偶性相同，于是不用担心逆序数的影响，DP 出的结果  $f_{r,0}$  ( $r$  是根) 即为答案。

时间复杂度  $O(n)$ ，期望得分 100，是本题的正解。

## 3 总结

### 3.1 思想总结

本题从 gcd 矩阵的思想出发，将做法拓展到树上。并利用树的性质，根据行列式问题中经常出现的正负抵消的技巧，抛弃大量无用计算，将问题转化成简洁的树形 DP 问题。

gcd 矩阵的思想还能拓展到其他特殊偏序集上的求交问题中。很多情况下，只需要利用该偏序集上的前缀和以及差分的性质，将矩阵表示成两个矩阵的积，就能将问题简单化。

### 3.2 出题过程

作者偶然解到了 gcd 矩阵，并发现树形结构也适合其思想。同时作者发现其不止在作为部分分的“ $A$  是  $1 \sim n$ ”简单拓展上有应用，在  $|A| < n$  的时候也有很多好的性质。在出题过程中很自然地想到使用 Cauchy-Binet 公式，同时受到“正负抵消”这样的技巧的启发，将问题变成了树形 DP 问题。

### 3.3 对题目的评价

本题作为难度适中的题目，代码量较小，细节量较少，有一定的区分度，具有一定的选拔意义。

由于该题目的出题过程和解题过程类似，题目的解法十分自然。本题没有涉及过难的知识，更考察了选手的思维。从题面的行列式计算到最终的树形 DP，若干 OI 题目和定理证明中常用的小技巧串联起来组成了这道题目，考察了选手的基本功。

为了照顾到部分作者没想到的解法，部分分被设计成  $O(n!)$ ,  $O(n^3)$ ,  $O(n^2)$ ,  $O(n)$  各个复杂度。作者也对部分分的思考难度估量了权重，保证良好的区分度。

综上，作者特意选择这道题为交流的试题，意在分享这道题巧妙解法。以及期待大家对这个模型的进一步拓展。