《Misaka Network》解题报告

重庆市南开中学 刘谨瑞

2025 年 10 月 31 日

1 题目

1.1 题目概述

给定两棵 n 个点的无根树 T_1, T_2，节点编号为 $1 \sim n$。

有一个 $1 \sim n$ 的排列 p，初始时 $p_i = i$，你可以进行若干次操作，每次操作选择两个点 u, v，满足 u, v 在 T_1 中相邻且 p_u, p_v 在 T_2 中相邻，然后交换 p_u 和 p_v，问能得到多少种不同的 p，答案对 998244353 取模。

t 组数据。

1.2 数据范围

令 $\sum n$ 表示一个测试点内所有数据的 n 的和。

对于所有数据，满足 $1 \le t \le 10^5, 1 \le n \le 10^6, \sum n \le 2 \times 10^6, 1 \le u, v \le n$，保证 T_1, T_2 是树。

子任务编号	n	$\sum n$	特殊限制	分值
1	≤ 10	≤ 100	无	4
2	$\le 10^5$	$\le 10^5$	$T_1 = T_2$	4
3			T_2 是一条链	12
4	≤ 100	≤ 1000	无	8
5	≤ 1000	≤ 9969		20
6	$\le 5 \times 10^4$	$\le 2 \times 10^5$		16
7	$\le 2 \times 10^5$	$\le 6 \times 10^5$		20
8	$\le 10^6$	$\le 2 \times 10^6$		16

2 解题过程

2.1 算法 1

直接搜索，可以通过子任务 1，期望得分 4 分。

2.2 算法 2

通过观察容易发现答案为匹配（不一定最大）的方案数，可以通过子任务 2，期望得分 4 分。

2.3 算法 3

见原题及其题解[1]，可以通过子任务 3，期望得分 12 分。

2.4 分析性质

上述做法没有任何优化的空间，考虑分析一些性质。

定义 1. 局面

定义一个局面为一个三元组 (T_1, T_2, p)，其中 T_1, T_2 均为 n 个点的无根树，p 为一个 $1 \sim n$ 的排列。

对于局面 $A = (T_1, T_2, p)$，定义 $\text{tree}_1(A) := T_1, \text{tree}_2(A) := T_2, \text{perm}(A) := p$。

定义 2. 局面的对称性

容易发现对于一个局面 (T_1, T_2, p)，T_1, T_2 的地位相同，我们称其为局面的对称性。考虑局面 (T_1, T_2, p) 和 (T_2, T_1, p^{-1}) 可以发现，在前者上执行的变换都能在后者找到一种对应的变换，同样也能得到，它们能通过一些变换得到的局面间也存在一一对应关系。

定义 3. 元变换

定义一个元变换为一个二元组 (u, v)，含义为交换 p_u, p_v。

为了方便，我们规定元变换和边是同一种元素。

定义 4. 变换

定义一个变换为若干个元变换组成的序列 $\{(u_1, v_1), \ldots, (u_k, v_k)\}$，含义为依次执行序列中的变换。

对于变换 $a = \{(u_1, v_1), \ldots, (u_k, v_k)\}$，称 $|a| = k$ 为 a 的长度，定义 $a_i = (u_i, v_i)$，定义 $a^{-1} = \{a_k, \ldots, a_1\}$ 为它的逆。

定义 5. 复合

定义变换 a, b 之间的复合 $a \circ b := \{a_1, \ldots, b_1, \ldots\}$，即拼接两个序列。

定义元变换 a 参与和其他元变换或变换之间的复合时，等价于变换 $\{a\}$ 参与这个复合。

容易证明元变换/变换之间的复合存在结合律。

定义 $\text{swap}(u, v)$ 为一个交换 u, v 的置换，即满足

$$\text{swap}(u, v)_i = \begin{cases} v & i = u \\ u & i = v \\ i & i \neq u \wedge i \neq v \end{cases}$$

定义局面和元变换的复合 $(T_1, T_2, p) \circ (u, v) := (T_1, T_2, \text{swap}(u, v) \circ p)$。

定义局面和变换的复合 $(T_1, T_2, p) \circ \{a_1, \ldots, a_k\} := (T_1, T_2, p) \circ a_1 \circ \cdots \circ a_k$。

[1] https://qoj.ac/problem/11531

定义 6. 复合的合法性

对于局面和元变换之间的复合 $(T_1, T_2, p) \circ (u, v)$，若 $(u, v) \in T_1 \wedge (p_u, p_v) \in T_2$，则我们称这个复合是合法的，否则称它是不合法的，并将其记为 $\text{valid}((T_1, T_2, p), (u, v))$。

对于局面和变换之间的复合 $(T_1, T_2, p) \circ \{a_1, \ldots, a_k\}$，若对于所有 $1 \le i \le k$，均满足 $\text{valid}((T_1, T_2, p) \circ \{a_1, \ldots, a_{i-1}\}, a_i)$，则我们称这个复合是合法的，否则称它是不合法的，并将其记为 $\text{valid}((T_1, T_2, p), \{a_1, \ldots, a_k\})$。

为了方便，我们规定若提到 $A \circ a$，则默认要求 $\text{valid}(A, a)$。

定义 7. 局面的可达性

对于两个局面 A, B，若存在一个变换 a 使得 $\text{valid}(A, a)$ 且 $A \circ a = B$，那么我们称 A 可达 B。由变换的可逆，可知可达性是双向的，即若 A 可达 B，那么 B 可达 A。

定理 1. *对于一个局面 A 和一个长度为 k 的变换 a，若满足 $\text{valid}(A, a), a_1 = a_k = (u, v)$，且 $\forall 1 < i < k, a_i \neq (u, v)$，那么 $\text{perm}(A)_u = \text{perm}(A \circ a)_u$ 且 $\text{perm}(A)_v = \text{perm}(A \circ a)_v$。*

证明. 对于 u，显然 $\text{perm}(A \circ a_1)_u, \ldots, \text{perm}(A \circ a_1 \circ \cdots \circ a_{k-1})_u$ 构成一条 T_2 上的路径，设路径上的边（带方向）构成的序列为 q_1，对于 v 同理，设为 q_2。因为 $a_1 = a_k = (u, v)$，所以 $(\text{perm}(A)_u, \text{perm}(A)_v), (\text{perm}(A \circ a)_u, \text{perm}(A \circ a)_v)$ 均为 T_2 中的边（同样带方向），分别设为 e_1, e_2。连接上述路径和边会得到一个 T_2 中的回路 $\langle e_1, q_1, \text{rev}(e_2), \text{rev}(q_2) \rangle$，因为 T_2 是树，不存在环，于是每条边后一定被经过偶数次且从两个方向经过的次数相同，又因为 $\forall 1 < i < k, a_i \neq (u, v)$，所以 q_1, q_2 中都不存在 e_1, e_2，那么只可能 $e_1 = e_2$，且方向相同（e_1 与 $\text{rev}(e_2)$ 方向相反），即 $\text{perm}(A)_u = \text{perm}(A \circ a)_u$ 且 $\text{perm}(A)_v = \text{perm}(A \circ a)_v$。

综上，原命题成立。 □

推论 1. *对于一个局面 A 和一个长度为 k 的变换 a，若满足 $\text{valid}(A, a)$，且 $a_1 = a_k = (u, v)$，那么 $\{\text{perm}(A)_u, \text{perm}(A)_v\} = \{\text{perm}(A \circ a)_u, \text{perm}(A \circ a)_v\}$。*

证明. 由 定理 1 显然。 □

推论 2. *对于一个局面 A，和两个长度分别为 k_a, k_b 变换 a, b，若满足 $\text{valid}(A, a) \wedge \text{valid}(A, b)$，且 $a_{k_a} = b_{k_b} = (u, v)$，那么 $\{\text{perm}(A \circ a)_u, \text{perm}(A \circ a)_v\} = \{\text{perm}(A \circ b)_u, \text{perm}(A \circ b)_v\}$。*

证明. 令 $A' = A \circ a, a' = a^{-1} \circ b$，由 推论 1 可知显然成立。 □

由 推论 2 可知，从一个初始局面开始，我们在任意一次进行变换 $(u, v) \in T$ 时，$\{p_u, p_v\}$ 都是一样的，也就是对应 T_2 中的一条边。又由 定义 2 局面的对称性可得，T_2 中的每一条边也对应 T_1 中的某一条边，即这种对应关系是双射。

定义 8. 对应关系

对于一个局面 $A = (T_1, T_2, p)$，对于 $e = (u, v) \in T_1$，若存在一个以 e 结尾的变换 a 使得 $\text{valid}(A, a)$，那么 $\{\text{perm}(A \circ a)_u, \text{perm}(A \circ a)_v\}$ 是确定的，且对应着 T_2 中的一条边，我们将其记作 $\text{match}(e)$ 或者 $\text{match}((u, v))$，对于 T_2 中的边同理。

现在假设我们能在足够低的时间复杂度内求出所有对应关系。

定义 9. 简单变换

对于一个变换 $a = \{a_1, \ldots, a_k\}$，若 $\forall 1 \le i < j \le k, a_i \neq a_j$，则称变换 a 是简单的，又称 a 是一个简单变换。

对于一个简单变换 a，定义 $\text{set}(a)$ 为 a 中所有元变换构成的集合。

引理 1. 对于一个局面 A 和一个长度为 k 的简单变换 a，若满足 $\text{valid}(A,a), a_1 = a_k = (u,v)$，且 $\forall 1 < i < k, a_i \neq (u,v)$，那么 $A \circ a = A \circ \{a_2, \ldots, a_{k-1}\}$。

证明. 设 $\text{perm}(A)_u = x, \text{perm}(A)_v = y$。首先，由 定理 1 可知 $\text{perm}(A)_u = \text{perm}(A \circ a)_u$ 且 $\text{perm}(A)_v = \text{perm}(A \circ a)_v$。对于 u，假设 a 中存在任何元变换 $a_i = (u,w) \neq (u,v)$，找到最小的满足这个条件的 i，那么显然 $\text{perm}(A \circ \{a_1, \ldots, a_i\})_w = y$，又因为 a 是简单变换，所以不存在 $j > i$ 使得 $a_j = a_i = (u,w)$，所以 $\text{perm}(A \circ \{a_1, \ldots, a_{k-1}\})_u$ 不可能为 y，矛盾，所以 a 中一定不存在 $a_i = (u,w) \neq (u,v)$。对于 v 同理。

既然 a 中不存在任何其他元变换影响 u 和 v，那么显然同时删去 a_1, a_k 对复合的结果也不会产生影响，于是原命题成立。 □

定理 2. 对于一个局面 A 和一个变换 a,若 $\text{valid}(A,a)$,则一定存在至少一个简单变换 b 满足 $\text{valid}(A,b)$ 且 $A \circ a = A \circ b$。

证明. 考虑数学归纳法。首先，若 $|a| \leq 1$，结论显然成立。若对于所有 $|a| = k-1$ 的 a，结论都成立，考虑 $|a| = k$ 的 $a = \{a_1, \ldots, a_k\}$。显然可以假设 $\{a_1, \ldots, a_{k-1}\}$ 为简单变换，因为若不满足条件，则可以将其替换为与之等价的简单变换。若 $\forall 1 \leq i < k, a_i \neq a_k$，那么它本身就是一个简单变换，否则一定存在恰好一个 i 使得 $a_i = a_k$。同时删去 a_i 和 a_k 后得到变换 b，显然 b 是简单变换，又由 引理 1 得 $A \circ a = A \circ b$，即原命题成立。 □

引理 2. 对于一个局面 A 和一个简单变换 a，若 $\text{valid}(a)$，那么对于所有 u，$\text{set}(a)$ 中所有与 u 相邻的边的对应边在 T_2 中构成一条以 p_u 为一个端点的链，且 $\text{perm}(A \circ a)_u$ 为这条链的另一个端点。

证明. 略。 □

定理 3. 对于一个局面 $A = (T_1, T_2, p)$ 和两个简单变换 a, b，若 $\text{valid}(A,a) \wedge \text{valid}(A,b)$，则 $\text{set}(a) = \text{set}(b) \Leftrightarrow A \circ a = A \circ b$。

证明. 若 $\text{set}(a) = \text{set}(b)$，考虑对于每个点 u，找到 $\text{set}(a)$ 中所有以它为其中一个端点的边，由 引理 2 得它们的对应边在 T_2 中一定构成一条以 p_u 为端点的链，且 $\text{perm}(A \circ a)_u$ 和 $\text{perm}(A \circ b)_u$ 都是这条链的另一个端点。既然对于所有点 u 都满足 $\text{perm}(A \circ a)_u = \text{perm}(A \circ b)_u$，那么显然 $\text{perm}(A \circ a) = \text{perm}(A \circ b)$，即 $\text{set}(a) = \text{set}(b) \Rightarrow A \circ a = A \circ b$。

若 $\text{set}(a) \neq \text{set}(b)$，所以一定存在一个 u 使得 $\text{set}(a)$ 和 $\text{set}(b)$ 中 u 的邻边构成的集合不同。又因为对应关系是双射，所以 $\text{set}(a)$ 和 $\text{set}(b)$ 中 u 的邻边的对应边构成的集合也不同。那么这些对应边在 T_2 中构成的链不同，由 引理 2，这两条链的一端一定是 p_u，所以一定有这两条链的另一端不同，否则它们就完全相同了，产生矛盾。而又由 引理 2 可知，$\text{perm}(A \circ a)_u$ 和 $\text{perm}(A \circ b)_u$ 分别为这两条链的另一端，所以 $\text{perm}(A \circ a)_u \neq \text{perm}(A \circ b)_u$，于是 $A \circ a \neq A \circ b$。即 $\neg(\text{set}(a) = \text{set}(b)) \Rightarrow \neg(A \circ a = A \circ b)$，也就是 $A \circ a = A \circ b \Rightarrow \text{set}(a) = \text{set}(b)$。

综上，原命题成立。 □

由 定理 2 和 定理 3，我们可以只钦定每条边是否在变换中出现。那么接下来的问题就是，如何判断对于一个钦定的边集 S，是否存在变换 a，使得 $\text{set}(a)$ 且 $\text{valid}(A,a)$。

定理 4. 对于一个局面 $A = (T_1, T_2, p)$ 和任意一个元变换构成的集合 S,满足对于所有 $a \in S$,都存在一个以它为结尾的变换 b 使得 $\text{valid}(A,b)$，那么，若对于所有 T_1 中的点 u，满足 $\{\text{match}((u,v)) : (u,v) \in T_1\}$ 在 T_2 上构成一条以 p_u 为端点的链，则一定存在一个简单变换 a 满足 $\text{set}(a) = S$，且 $\text{valid}(A,a)$。

证明. 略。 □

于是我们得到了一个跟最终局面一一对应的状态，而且只有 T_1 上相邻两个点的信息会互相影响。

2.5 算法 4

根据上面分析的性质，考虑在 T_1 上进行树形 DP，设 $f_{u,0/1}$ 表示在 u 到父亲 fa_u 的边是/否在变换中出现过的情况下，u 子树内在变换中的边构成的不同集合的数量。

对于一个点 u，遍历所有 T_1 中的邻边 $\{(u,v)\}$，考虑 $\{\text{match}((u,v))\}$ 在 T_2 中构成的子图，枚举子图中与 p_u 连通的点 x，那么 $p_u \rightsquigarrow x$ 路径上的所有边在 T_1 中的对应边都是在变换中出现过的，其他边则一定没有。这时对于 u 的每个儿子 v，若 (u,v) 出现过，则贡献为 $f_{v,1}$，否则为 $f_{v,0}$。将所有儿子的贡献相乘，若 (u,fa_u) 出现过，则贡献到 $f_{u,1}$，否则贡献到 $f_{u,0}$。特别地，若 u 为根，则一定贡献到 $f_{u,0}$。

令 1 为根，则最终答案为 $f_{1,0}$。

精细实现可以做到时间复杂度 $O(n)$，所以这部分不是本题的时间复杂度瓶颈。

2.6 算法 5

现在的问题变为，对于一个局面 $A=(T_1,T_2,p)$，如何求出所有对应关系。

考虑如何判断 T_1 中的边 (u,v) 和 T_2 中的边 (x,y) 是互相对应的。

令 $u'=p^{-1}(x),v'=p^{-1}(y)$，若 u',v' 在删去 T_1 中边 (u,v) 后，位于同一个连通块内，那么显然 (u,v) 和 (x,y) 一定不是对应的。不妨假设删去边 (u,v) 后，u',v' 分别和 u,v 在同一个连通块内。

那么我们要判断的就是是否存在一个变换 a 使得 $\text{perm}(A\circ a)_u=x$ 且 $\text{perm}(A\circ a)_v=y$。

我们假设 a 是满足条件的变换中长度最短的，或者不存在 a 的一个前缀满足这个条件。

首先，若存在满足条件的变换也一定存在满足我们假设的变换。其次，我们可以知道 a 中不会出现 (u,v) 这个元变换，因为若存在 a，那么 (u,v) 对应 (x,y)，于是在删去 a 中这个元变换及之后的元变换后的变换也是满足条件的，与我们的假设矛盾。

于是两侧独立，那么我们只需要考虑是否存在 a 使得 $\text{perm}(A\circ a)_u=x$，另一边同理。

考虑 u' 到 u 路径上的点，假设依次为 $r_1=u',\ldots,r_k=u$。显然我们需要 $p_{r_1},\ldots,p_{r_k}$ 依次变为 x，所以对于每个 $1\le i<k$，(r_i,r_{i+1}) 一定对应 T_2 中一条以 x 为端点的边，假设为 (x,s_i)。因为所有的 (r_i,r_{i+1}) 都已经确定有对应的边了，所以我们将所有 $p_{r_{i+1}}$ 变为 s_i 的过程是独立的，也就是说我们把判断变成了更小的子问题，即只考虑某个连通块内的元变换，是否存在一个它们构成的变换 b 使得 $\text{perm}(A\circ a)_{r_{i+1}}=s_i$。

直接做显然是指数级的，因为要记录连通块信息，所以需要考虑一些更优美的实现方式。

因为对应关系只有不超过 $n-1$ 对，所以枚举 $(n-1)^2$ 对去判断是很浪费的，考虑以一种类似增广的方式去找到所有对应关系。

我们对于 T_1 中每个点 u 维护一个集合 $S_u=\{\text{perm}(A\circ a)_u:\text{valid}(A,a)\}$，即通过合法的变换可以使得 p_u 变为哪些数。在算法最开始，令 $S_u=\emptyset$，然后按 $u=1,\ldots,n$ 依次令 $S_u\leftarrow S_u\cup\{p_u\}$。在令 $S_u\leftarrow S_u\cup\{x\}$ 的时候，枚举 $(u,v)\in T_1,y\in S_v$，若 $(x,y)\in T_2$，且 T_1 中的 (u,v) 和 T_2 中的 (x,y) 都还没有找到对应边，那么 (u,v) 和 (x,y) 便是一对对应边。找到 (u,v) 和 (x,y) 互相对应后，令 $S_u\leftarrow S_u\cup\{y\},S_v=\leftarrow S_v\{x\}$ 并递归地执行上面操作。

将上述做法改为类似广搜的实现也是正确的。

首先容易发现这样做不可能漏掉任何对应关系，考虑为什么它也不会找到不合法的对应关系。其实重点在于 (u,v) 和 (x,y) 在之前没有找到过对应边才会认为它们是互相对应的。

考虑之前枚举 $(u,v)\in T_1,(x,y)\in T_2$ 再判断的条件。首先是要求 (u,v) 在之前没有找到对应边，这个显然是满足的。然后分别考虑两侧，两侧的情况本质相同，所以我们只需要说明其中一边是正确的。

按照算法流程，我们显然是会依次把 x 加入 $S_{r_1},\dots,S_{r_k}$。在此之前我们会将 s_i 加入 $S_{r_{i+1}}$，在此过程中，一定不会有 (r_i,r_{i+1}) 这些变换，所以也满足每一部分独立这个条件。

最后考虑递归到子问题，因为问题规模是严格减小的，而且这些问题都是本质相同的，所以通过归纳法容易证明正确性。

直接实现可以做到 $O(n^2)$ 时间复杂度，结合算法 4 可以通过子任务 1,4,5，期望得分 32 分。如果再结合算法 2,3，则可以获得 48 分。

2.7 算法 6

考虑直接优化算法 5。

令 $S'_u=\{y:x\in S_u,(x,y)\in T_2\}$，考虑对所有 u 维护 S'_u。建立一个 $n\times n$ 的网格，称其第 u 行 x 列的格子为 (u,x)，表示 $[x\in S'_u]$。最开始所有格子都为 0。

对于 $S_u\leftarrow x$，我们会对于所有 $(x,y)\in T_2$，将 (u,y) 设为 1，然后对于所有 $(u,v)\in T_1$，查询 (v,x) 是否为 1，如果为 1，那么我们就可能找到了一对对应关系。

考虑求出 T_1,T_2 的 bfs 序，一个点的邻域在 bfs 中最多分为一个区间和一个单点，于是操作变为行区间修改和列区间查询。又因为操作次数显然为 $O(n)$ 次，也容易证明查询到 1 但是不是新的对应关系的情况也只有 $O(n)$ 次。所以我们可以直接使用分块或者树套树做到 $O(n\sqrt{n})$ 或 $O(n\log^2 n)$ 的时间复杂度，结合算法 4 可以通过子任务 1-6，期望得分 64 分。

2.8 算法 7

考虑若知道一组几乎正确的对应关系（即它包含所有正确的对应关系，但是可能多了一些错误的对应关系），假设一共 m 对，如何求出哪些是正确的。

将算法 5 改为只枚举可能正确的对应关系，用哈希表这类数据结构维护，可以做到时间复杂度 $O(n+m)$，在 $O(m)=O(n)$ 的情况下即为 $O(n)$。

2.9 算法 8

对于局面 $A=(T_1,T_2,p)$，枚举 T_2 中的边 (x,y)，令 $u=p^{-1}(x),v=p^{-1}(y)$，易证 (x,y) 的对应边只可能在 T_1 中 $u\rightsquigarrow v$ 这条路径上。设 $u\rightsquigarrow v$ 依次经过 $r_1=u,\dots,r_k=v$ 这些点，若 $\mathrm{match}((x,y))=(r_i,r_{i+1})$，那么与之前类似地，独立地考虑两侧，若存在两个均不包含 (r_i,r_{i+1}) 的变换 a_1,a_2 使得 $\mathrm{perm}(A\circ a_1)_{r_i}=x$ 且 $\mathrm{perm}(A\circ a_2)_{r_{i+1}}=y$，则它们互相对应。

考虑放宽限制，对于 $\mathrm{perm}(A\circ a_1)_{r_i}=x$，显然有，在 T_2 上，$p_{r_1},\dots,p_{r_i}$ 与 x 之间的路径不经过 y，$p_{r_{i+1}},\dots,p_{r_k}$ 同理。即，断开 T_2 中的边 (x,y) 后，$p_{r_1},\dots,p_{r_i}$ 与 x 在同一个连通块，$p_{r_{i+1}},\dots,p_{r_k}$ 与 y 在同一个连通块。一个很好的性质是，在现新的限制下，(x,y) 依然只可能对应一条边。

如果直接实现，可以做到时间复杂度为 $O(n\log^2 n)$。具体来说，首先线段树合并可以维护子树的并，然后可以先树剖，将链转化为 $O(\log n)$ 个 dfs 区间，再在线段树上二分，也可以将单点加链和转化为区间加单点查，二分时在线段树上查询链和。这样做相较于之前的做法唯一的优势是空间复杂度更小，但是时间复杂度没有变化。

考虑继续优化，但是在现在的限制下很难做到更优时间复杂度了，于是转而继续放宽限制。现在我们只要求存在 a_1 或者存在 a_2，显然这样还是只有至多一条可能的对应边。

不妨设，在 T_2 中 y 为 x 的父亲，再将限制变为，断开 T_2 中的边 (x,y) 后，$p_{r_1},\dots,p_{r_i}$ 与 x 在同一个连通块。这个限制等价于 $p_{r_1},\dots,p_{r_i}$ 都在 T_2 中 x 的子树内，即一个 dfs 区间内。

求出 T_2 的 dfs 序后在 T_1 上使用树剖或者倍增都能在 $O(n\log n)$ 的时间复杂度内求出对应边。结合算法 4,7 可以通过子任务 1-7，期望得分 84 分。

2.10 算法 9

考虑在 T_2 中从叶子的邻边开始从下到上确定它们的对应边。对于叶子 x，考虑边 (x, fa_x)，其对应边显然只可能为 T_1 中 $p^{-1}(x) \rightsquigarrow p^{-1}(fa_x)$ 路径上的第一条边。同样地，对于任意一个点 x，边 (x, fa_x) 的对应边只可能为 T_1 中 $p^{-1}(x) \rightsquigarrow p^{-1}(fa_x)$ 路径上的第一条不是 x 与 x 儿子间的边的对应边的边，直接暴力地找这条边，最多会跳过儿子数这么多边，最后等价于 x 为叶子的情况。

时间复杂度瓶颈在于求一个点在另一个点的哪个儿子的子树内，求出 dfs 序然后在后者的儿子中二分，容易做到 $O(n\log n)$。可以用线性 k 级祖先优化到 $O(n)$，但是没有必要。

对于 $p^{-1}(fa_x)$ 在 T_1 中是 $p^{-1}(x)$ 的后代的情况，改为在处理 fa_x 时向上寻找，精细实现同样可以做到 $O(n)$。

结合算法 4,7 可以通过所有子任务，期望得分 100 分。

2.11 关于算法 7

对于通过算法 8,9 找到的可能正确的对应关系，跳过算法 7 直接执行算法 4，似乎也能得到正确的答案，笔者目前找不到反例但是也不会证明。

但是需要注意，在这种情况下，可能需要对算法 4 进行一些细节上的修改。