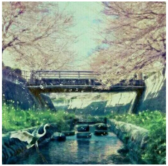
NOI 2023
桂花树 (osmanthus) 解题报告

ltst
Special Thanks: unzcjouhi / Elegia / liuzhangfeiabc

2023 年 7 月 22 日



简要题意

给定一棵 n 个节点的以 1 为根的树 T 和整数 k，求有多少个 $n+m$ 个节点的以 1 为根的树 T'，使得

- $\forall 1 \le i,j \le n, LCA_T(i,j) = LCA_{T'}(i,j)$;
- $\forall 1 \le i,j \le n+m, LCA_{T'}(i,j) \le \max(i,j)+k$。

答案对 (10^9+7) 取模。$n \le 3\times 10^4, m \le 3000, 0 \le k \le 10$

吐槽

~~你这棵树怎么输入进来没用啊？~~
~~怎么标算这么菜啊？~~
~~这题真没超纲吗？~~
……

$n, m \leq 4$

暴力枚举所有树形态判断是否满足题目条件，由于范围非常小所以不管复杂度是啥应该都能过，获得 10 分。

$m = 0$

输出 1，获得额外的 5 分。

$m=1$

节点 $n+1$ 有两种“放入” T 的方式：插入在其中一条边里（方案数 $n-1$），或者接在某个节点下面（方案数 n）。
输出 $(2n-1)$ 获得额外的 10 分。

$m=2$

跟 $m=1$ 类似，讨论两个节点怎么放入 T 里。注意可以 $(n+2)$ 接在 $(n+1)$ 下面，或者 $(n+1)$ 接在 $(n+2)$ 下面。

$m=2$

跟 $m=1$ 类似，讨论两个节点怎么放入 T 里。注意可以 $(n+2)$ 接在 $(n+1)$ 下面，或者 $(n+1)$ 接在 $(n+2)$ 下面。
对于 $k=0$ 的情况，需要注意 $(n+1)$ 接在 $(n+2)$ 下面且 $(n+2)$ 插在一条边里的方案非法。
出题人贴心地将 $k=0$ 和 $k \neq 0$ 的情况都给在了大样例里。

$m=2$

跟 $m=1$ 类似，讨论两个节点怎么放入 T 里。注意可以 $(n+2)$ 接在 $(n+1)$ 下面，或者 $(n+1)$ 接在 $(n+2)$ 下面。
对于 $k=0$ 的情况，需要注意 $(n+1)$ 接在 $(n+2)$ 下面且 $(n+2)$ 插在一条边里的方案非法。
出题人贴心地将 $k=0$ 和 $k\neq 0$ 的情况都给在了大样例里。
输出

$$\begin{cases}(2n-1)(2n+1), & k=0\\(2n-1)(2n+1)+(n-1), & k>0\end{cases}$$

获得额外的 10 分。

$n=1, k=0, m \leq 100$

$n=1$ 意味着 LCA 一致的条件是没用的，我们需要考虑的是所有有根树。
同时 $k=0$ 意味着第二个条件（$LCA(i,j) \leq \max(i,j)$）只关注节点编号的大小关系，而不关注它们的具体数值。

$n=1, k=0, m \le 100$

$n=1$ 意味着 LCA 一致的条件是没用的，我们需要考虑的是所有有根树。
同时 $k=0$ 意味着第二个条件（$LCA(i,j) \le \max(i,j)$）只关注节点编号的大小关系，而不关注它们的具体数值。
设 f_n 表示将 $[n]=\{1,2,\cdots,n\}$ 组成一棵（可以以任意节点为根的）有根树使得其满足第二个条件的方案数，g_n 表示将 $[n]$ 组成一个满足第二个条件的森林（$LCA(i,j)$ 只算在同一棵树里）的方案数，那么答案就是 g_m。

$n = 1, k = 0, m \leq 100$

g_n 的转移比较容易：每次我们选一个子集把它分出去作为一棵新的子树即可。注意我们提到的 $k = 0$ 的条件，意味着任选这个子集，它们构成合法的树的方案数都是 f_x，剩余部分都是 g_{n-x}。为了保证顺序唯一我们需要保证子集里的最小值必须要在枚举出来的这个子树里。

$$g_n = \sum_{i=1}^{n} \binom{n-1}{i-1} f_i g_{n-i}.$$

$n = 1, k = 0, m \leq 100$

对于 f_n，我们先枚举根的编号。如果根不是 1，那么比根编号小的必须要在同一棵子树里。枚举这棵子树的大小，有

$$f_n = g_{n-1} + \sum_{i=2}^{n} \sum_{j=i-1}^{n-1} \binom{n-i}{j-(i-1)} f_j g_{n-j-1}.$$

复杂度 $O(m^3)$，可以获得额外的 10 分。

$n = 1, k = 0, m \le 3000$

注意到如果根不是 1，那么枚举了 1 所在的子树包含哪些节点之后，根节点直接确定了。所以有

$$f_n = g_{n-1} + \sum_{i=1}^{n-1} \binom{n-1}{i-1} f_i g_{n-i-1}.$$

当然你也可以通过交换 $O(m^3)$ 做法的求和顺序然后用 $\sum_{i=0}^{m} \binom{n+i}{i} = \binom{n+m+1}{n+1}$ 得到同样做法。
复杂度 $O(m^2)$，可以获得额外的 5 分。

$n=1, k=0, m \le 3000$

注意到如果根不是 1，那么枚举了 1 所在的子树包含哪些节点之后，根节点直接确定了。所以有

$$f_n = g_{n-1} + \sum_{i=1}^{n-1} \binom{n-1}{i-1} f_i g_{n-i-1}.$$

当然你也可以通过交换 $O(m^3)$ 做法的求和顺序然后用 $\sum_{i=0}^{m} \binom{n+i}{i} = \binom{n+m+1}{n+1}$ 得到同样做法。
复杂度 $O(m^2)$，可以获得额外的 5 分。
你也可以用 $O(m^3)$ 程序打表获得额外的 5 分。

$k=0, n, m \le 100$

最后 T' 一定是由 T 按照这样的方法得来：将 m 个新节点划分给每条边和每个节点，每个节点划分到的新节点挂在其底下，每条边分到的节点作为这条边扩展出的新节点。
由于每条边和每个点的方案数只和分过来的新节点数相关，所以可以发现这是一个背包。
每个点的方案数就是 $g_\star$，每条边的方案数可以类似上面的方法计算，需要注意根节点有一棵子树连向比新加节点编号都更小的节点。
复杂度 $O(nm^2+m^3)$，可以获得额外的 10 分。

$k=0$

其实答案有很好的形式:

$$(2n-1)(2n+1)\cdots(2n+2m-3).$$

接下来我们来证明它。

$k=0$

其实答案有很好的形式：

$$(2n-1)(2n+1)\cdots(2n+2m-3).$$

接下来我们来证明它。
对于一个满足条件的树，考察 $(n+m)$，其度数必须不超过 2，否则第二个条件无法满足。如果 $(n+m)$ 是叶子就把它删掉，如果度数为 2 就把两条邻边缩起来，这样得到了一棵 $(n+m-1)$ 个节点的合法的树。

$k = 0$

其实答案有很好的形式：

$$(2n-1)(2n+1)\cdots(2n+2m-3).$$

接下来我们来证明它。
对于一个满足条件的树，考察 $(n+m)$，其度数必须不超过 2，否则第二个条件无法满足。如果 $(n+m)$ 是叶子就把它删掉，如果度数为 2 就把两条邻边缩起来，这样得到了一棵 $(n+m-1)$ 个节点的合法的树。
同时对于 $(n+m-1)$ 个节点的合法的树，插在任何一条边里或挂在任何一个节点底下都能得到一棵 $(n+m)$ 个节点的合法的树。于是这样归纳下去即可证明。

$k = 0$

其实答案有很好的形式：

$$(2n-1)(2n+1)\cdots(2n+2m-3).$$

接下来我们来证明它。
对于一个满足条件的树，考察 $(n+m)$，其度数必须不超过 2，否则第二个条件无法满足。如果 $(n+m)$ 是叶子就把它删掉，如果度数为 2 就把两条邻边缩起来，这样得到了一棵 $(n+m-1)$ 个节点的合法的树。
同时对于 $(n+m-1)$ 个节点的合法的树，插在任何一条边里或挂在任何一个节点底下都能得到一棵 $(n+m)$ 个节点的合法的树。于是这样归纳下去即可证明。
可以获得额外的 10 分，而只写这个式子就可以直接获得 50 分，拼上 $m=2$ 和大暴力就可以获得 70 分，真是太良心了！ !1

用同样的思路做一下 $k>0$？

仍然考虑 $(n+m)$，此时其可以接多个儿子，但题目条件告诉我们，至多只有一棵子树的编号不落在 $[n+m-k, n+m-1]$ 里。

用同样的思路做一下 $k>0$？

仍然考虑 $(n+m)$，此时其可以接多个儿子，但题目条件告诉我们，至多只有一棵子树的编号不落在 $[n+m-k, n+m-1]$ 里。考虑删除 $(n+m)$ 的时候把编号完全落在 $[n+m-k, n+m-1]$ 的子树全删了，这样删掉之后剩下的树（不改变剩下节点的编号）还是合法的。

用同样的思路做一下 $k>0$？

仍然考虑 $(n+m)$，此时其可以接多个儿子，但题目条件告诉我们，至多只有一棵子树的编号不落在 $[n+m-k, n+m-1]$ 里。
考虑删除 $(n+m)$ 的时候把编号完全落在 $[n+m-k, n+m-1]$ 的子树全删了，这样删掉之后剩下的树（不改变剩下节点的编号）还是合法的。
枚举这个编号完全落在子树里的子集 S，由于 S 内部不受条件约束，方案数完全由 $|S|$ 决定，可以预处理求出。分别转移 $(n+m)$ 没有 $< n+m-k$ 的子树（也就是剩余部分的一个叶子）或者恰有一个（也就是从剩余部分的一条边上插进去）的情况。

用同样的思路做一下 $k>0$？

仍然考虑 $(n+m)$，此时其可以接多个儿子，但题目条件告诉我们，至多只有一棵子树的编号不落在 $[n+m-k, n+m-1]$ 里。考虑删除 $(n+m)$ 的时候把编号完全落在 $[n+m-k, n+m-1]$ 的子树全删了，这样删掉之后剩下的树（不改变剩下节点的编号）还是合法的。

枚举这个编号完全落在子树里的子集 S，由于 S 内部不受条件约束，方案数完全由 $|S|$ 决定，可以预处理求出。分别转移 $(n+m)$ 没有 $< n+m-k$ 的子树（也就是剩余部分的一个叶子）或者恰有一个（也就是从剩余部分的一条边上插进去）的情况。

注意恰有一个的情况实际上算的是 ≤ 1 个的情况，而且会把没有的情况算多次。所以还需要容斥一下。

用同样的思路做一下 $k>0$？

仍然考虑 $(n+m)$，此时其可以接多个儿子，但题目条件告诉我们，至多只有一棵子树的编号不落在 $[n+m-k, n+m-1]$ 里。
考虑删除 $(n+m)$ 的时候把编号完全落在 $[n+m-k, n+m-1]$ 的子树全删了，这样删掉之后剩下的树（不改变剩下节点的编号）还是合法的。
枚举这个编号完全落在子树里的子集 S，由于 S 内部不受条件约束，方案数完全由 $|S|$ 决定，可以预处理求出。分别转移 $(n+m)$ 没有 $<n+m-k$ 的子树（也就是剩余部分的一个叶子）或者恰有一个（也就是从剩余部分的一条边上插进去）的情况。
注意恰有一个的情况实际上算的是 ≤ 1 个的情况，而且会把没有的情况算多次。所以还需要容斥一下。
状压 $[n+m-k, n+m-1]$ 是否被删除，每一次转移枚举子集或高维前缀和并额外记录子集大小。复杂度 $O(m3^k)$ 或 $O(m2^kk^2)$，期望获得 85 分。

思路 1 - 条件转化

对于一棵树是否满足第二个条件，我们关心的是每个点的所有儿子的子树编号 $\min$ 的次小值。当这个次小值不小于当前节点 $-k$ 的时候这个点就是合法的。

思路 1 - 条件转化

对于一棵树是否满足第二个条件，我们关心的是每个点的所有儿子的子树编号 $\min$ 的次小值。当这个次小值不小于当前节点 $-k$ 的时候这个点就是合法的。
考虑一个链剖的结构：对于每个节点 x，设其子树最小编号为 $\min(x)$。若 $\min(x) < x$，且 $\min(x)$ 在其儿子 y 的子树里，就将 $x \to y$ 的边标为重边，其余边标为轻边；如果 $\min(x) = x$ 则所有儿子边都标为轻边。

思路 1 - 条件转化

对于一棵树是否满足第二个条件，我们关心的是每个点的所有儿子的子树编号 $\min$ 的次小值。当这个次小值不小于当前节点 $-k$ 的时候这个点就是合法的。

考虑一个链剖的结构：对于每个节点 x，设其子树最小编号为 $\min(x)$。若 $\min(x) < x$，且 $\min(x)$ 在其儿子 y 的子树里，就将 $x \to y$ 的边标为重边，其余边标为轻边；如果 $\min(x) = x$ 则所有儿子边都标为轻边。

这样我们得到了这棵树的一个链剖分，且对于第二个条件，我们可以转化为：对于每个节点 x 以及一条轻边 $x \to y$，我们需要 y 的重链底的编号 $\geq x - k$，即对于每条重链 $p \rightsquigarrow q$，我们需要 $p \geq fa_q - k$。

思路 1 - $k=0$

我们首先考虑 $k=0$ 的情况。由于初始 $fa_i < i$，这棵树的链剖是若干个单点。

我们考虑按照编号顺序依次插入每个点，并维护链剖形态。每次新加入的点有两种行为：

- 插在某个链里面，方案数为 $n-1$（对应插入一条边中）;
- 新增一条链让当前节点作为链底，此时这条链的链顶父亲编号必须要小于当前节点编号，方案数为 n（对应作为一个叶子）。

这可以无重无漏地枚举出所有满足条件的树，和上面 $k=0$ 的证明是一样的。

思路 1 - $k>0$

考虑 $k>0$，此时新增链的链顶父亲可以大于当前节点编号。
设当前加入的点是 p，新增链的链顶父亲是 $p+\delta$，那么，$p+\delta$ 的子树 $\min$ 小于 p。
所以我们先把 $p+\delta$ 插进当前某条重链之后，让 p 作为其儿子。

思路 1 - $k>0$

考虑 $k>0$，此时新增链的链顶父亲可以大于当前节点编号。
设当前加入的点是 p，新增链的链顶父亲是 $p+\delta$，那么，$p+\delta$ 的子树 $\min$ 小于 p。
所以我们先把 $p+\delta$ 插进当前某条重链之后，让 p 作为其儿子。
因此对于新的点，现在的选择是：

- 插在某个链里面；
- 新增一条链让当前节点作为链底，其链顶的父亲是已经存在的节点；
- 新增一条链让当前节点作为链底，其链顶的父亲是不存在在其中的节点，此时我们先把这个节点插进去之后再让当前节点作为其儿子。

思路 1 - $k>0$

注意到由于加 p 时当前已经加入的节点编号不超过 $(p+k)$ 所以第二步可以任选一个已经加入的节点作为父亲。

思路 1 - $k>0$

注意到由于加 p 时当前已经加入的节点编号不超过 $(p+k)$ 所以第二步可以任选一个已经加入的节点作为父亲。
按照上面的方法可以得到一棵合法的树，且每棵合法的树恰有一种方法得到。我们可以用类似 $k=0$ 的证明，如果 $(n+m)$ 是由第三种情况加入的，设其子树次小 $\min$ 为 $n+m-k$，那么把 $n+m$ 绑在 $n+m-k$ 上然后在 $n+m-k$ 被删掉的时候（此时 $n+m$ 度数为 2 ）处理掉。

思路 1 - $k > 0$

注意到由于加 p 时当前已经加入的节点编号不超过 $(p+k)$ 所以第二步可以任选一个已经加入的节点作为父亲。
按照上面的方法可以得到一棵合法的树，且每棵合法的树恰有一种方法得到。我们可以用类似 $k=0$ 的证明，如果 $(n+m)$ 是由第三种情况加入的，设其子树次小 $\min$ 为 $n+m-k$，那么把 $n+m$ 绑在 $n+m-k$ 上然后在 $n+m-k$ 被删掉的时候（此时 $n+m$ 度数为 2 ）处理掉。
插入 p 时用状压维护 $(p+1)\sim(p+k)$ 中哪些点被提前插进去就可以维护三种选择的方案数，复杂度 $O(m2^kk)$，可以通过所有测试点。

思路 2 - 只考虑第一个条件

我们先忽略掉第二个条件。注意到第一个条件其实说的是 T' 在 $1 \sim n$ 里虚树是 T。

思路 2 - 只考虑第一个条件

我们先忽略掉第二个条件。注意到第一个条件其实说的是 T' 在 $1 \sim n$ 里虚树是 T。

我们考虑从小到大加点并维护虚树。具体地，考察如下过程：

1. 维护若干个点在 T' 中构成的虚树 S，最初 $S = T$。
2. 找到编号最小的还未加入 S 中的节点 x。
3. 给 x 决定一种插入 S 中的方式。这个写在下面。
4. 若 S 的节点数不是 $(n+m)$，回到步骤 2，否则 S 是一棵以 T 为主干的树。

第三步有如下几种方式：

1. 选择某个已经在 S 里的节点，作为其直接儿子；
2. 选择某条 S 中的边 (u,v)，断开它并连上 $(u,x)(x,v)$。
3. 再选择某个不在 S 中的节点 y，选择某条 S 中的边 (u,v)，断开它并连上 $(u,y)(x,y)(v,y)$。

思路 2 - 只考虑第一个条件

我们先忽略掉第二个条件。注意到第一个条件其实说的是 T' 在 $1 \sim n$ 里虚树是 T。

我们考虑从小到大加点并维护虚树。具体地，考察如下过程：

1. 维护若干个点在 T' 中构成的虚树 S，最初 $S = T$。
2. 找到编号最小的还未加入 S 中的节点 x。
3. 给 x 决定一种插入 S 中的方式。这个写在下面。
4. 若 S 的节点数不是 $(n+m)$，回到步骤 2，否则 S 是一棵以 T 为主干的树。

第三步有如下几种方式：

1. 选择某个已经在 S 里的节点，作为其直接儿子；
2. 选择某条 S 中的边 (u, v)，断开它并连上 $(u, x)(x, v)$。
3. 再选择某个不在 S 中的节点 y，选择某条 S 中的边 (u, v)，断开它并连上 $(u, y)(x, y)(v, y)$。

这相当于按照从小到大的顺序枚举了虚树形态，故无重地生成所有以 T 为主干的树。

思路 2 - 现在考虑第二个条件

在有 $LCA(i,j) \le \max(i,j)+k$ 的限制下，对于步骤 3，我们要求 $y \le x+k$。我们接下来说明，只需要满足这一个选 y 的条件，那么我们任意选择上面的插入的方式，都可以获得一个合法的树。

思路 2 - 现在考虑第二个条件

在有 $LCA(i,j) \le \max(i,j) + k$ 的限制下，对于步骤 3，我们要求 $y \le x+k$。我们接下来说明，只需要满足这一个选 y 的条件，那么我们任意选择上面的插入的方式，都可以获得一个合法的树。具体地，我们发现在这样的约束下，在第二步找出 x 后，S 中最大的点编号为 $x+k-1$，且叶子节点的编号一定都小于 x。依次考虑所有三种操作：

1. 由于 S 中最大的点编号是 $x+k-1$，加入 x 引入的 $LCA(x,\cdot)$ 一定不超过 $x+k-1$，所以一定合法；
2. 由于是加在一条边中间，所以不会有 $LCA(i,j)$ 满足 $i \ne x, j \ne x, LCA(i,j) = x$，所以一定合法；
3. 这个则是 1 和 2 的结合。我们会有一个问题是，y 是否有可能可以放大于 $x+k$ 的元素，这由 “叶子节点编号小于 x” 可以得出是不行的。

思路 2 - 现在考虑第二个条件

在有 $LCA(i,j) \le \max(i,j)+k$ 的限制下，对于步骤 3，我们要求 $y \le x+k$。我们接下来说明，只需要满足这一个选 y 的条件，那么我们任意选择上面的插入的方式，都可以获得一个合法的树。具体地，我们发现在这样的约束下，在第二步找出 x 后，S 中最大的点编号为 $x+k-1$，且叶子节点的编号一定都小于 x。依次考虑所有三种操作：

1. 由于 S 中最大的点编号是 $x+k-1$，加入 x 引入的 $LCA(x,\cdot)$ 一定不超过 $x+k-1$，所以一定合法；
2. 由于是加在一条边中间，所以不会有 $LCA(i,j)$ 满足 $i \ne x, j \ne x, LCA(i,j)=x$，所以一定合法；
3. 这个则是 1 和 2 的结合。我们会有一个问题是，y 是否有可能可以放大于 $x+k$ 的元素，这由 "叶子节点编号小于 x" 可以得出是不行的。

这样我们从小到大考虑每个点，状压维护后面 k 个点是否加入了 S 里即可。复杂度 $O(m2^k k)$，可以通过。

End

Thanks for your listening!