TreeDistance 解题报告

绍兴一中 张恒捷

1 试题来源

Topcoder TCO2014 Round 3B 1000pt

2 试题大意

给出一棵 N 个点的带标号无根树。每一次操作可以选择一条树上的边将其删除，然后再选择两个点连接起来，要保证连接起来以后仍然是一棵树。你现在最多可以操作 K 次，求能生成多少种不同的树。当存在一对边 (u, v) 使得一棵树中有 (u, v) 边，另一棵树没有时，认为两棵树不同。答案模 $10^9 + 7$ 后输出。

数据范围：$N, K \leq 50$

3 算法介绍

定理. 两棵 N 个点的树 P, Q，假设它们有 k 条边是共同拥有的，则 P 最少经过 $N-1-k$ 次操作就能变成 Q。

Proof. 不妨设 P 最少需要 t 次操作才能变成 Q。很显然每次操作最多加入一条本不属于 P 的边，所以有 $t \geq N-1-k$。每次操作可以把一条在 Q 中不在 P 中的边选作待加入的边，不妨设该边为 (x, y)，那么肯定存在一条在 P 中 x 到 y 的路径上，且不在 Q 中的边，否则 Q 中就有环了。使用这两条边做一次操作可以使 k 减少1。因此可以得到 $t \leq N-1-k$，所以 $t = N-1-k$，原命题得证。 □

根据定理，可以发现题目要求的就是：给出一棵树 T，求出拥有至少 $N-1-k$ 条 T 中边的树个数。这个问题可以用容斥来做。设 $g(i)$ 为强制选择 T 中 i 条边后能组成树的方案数。但是怎么来算它呢？

考虑现在已经选了某 i 条边，要算组成树的方案可以先把这些边缩起来，然后用基尔霍夫矩阵来求。假设缩完边之后每个点分别由 $a_1, a_2 \ldots, a_m$ 个点缩成的，那么第 i 个点与第 j 个点间的边数为 $a_i \times a_j$。这里列出矩阵：

$$\mathbf{A} = \begin{pmatrix} a_1 \sum\limits_{i \neq 1} a_i & -a_1 a_2 & -a_1 a_3 & \cdots & -a_1 a_{n-1} \\ -a_2 a_1 & a_2 \sum\limits_{i \neq 2} a_i & -a_2 a_3 & \cdots & -a_2 a_{n-1} \\ -a_3 a_1 & -a_3 a_2 & a_3 \sum\limits_{i \neq 3} a_i & \cdots & -a_3 a_{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n-1} a_1 & -a_{n-1} a_2 & -a_{n-1} a_3 & \cdots & a_{n-1} \sum\limits_{i \neq n-1} a_i \end{pmatrix}$$

现在要求出 A 的行列式，将第一行加上其他所有行，行列式值不变。

$$det(\mathbf{A}) = \begin{pmatrix} a_1 a_n & a_2 a_n & a_3 a_n & \cdots & a_{n-1} a_n \\ -a_2 a_1 & a_2 \sum\limits_{i \neq 2} a_i & -a_2 a_3 & \cdots & -a_2 a_{n-1} \\ -a_3 a_1 & -a_3 a_2 & a_3 \sum\limits_{i \neq 3} a_i & \cdots & -a_3 a_{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n-1} a_1 & -a_{n-1} a_2 & -a_{n-1} a_3 & \cdots & a_{n-1} \sum\limits_{i \neq n-1} a_i \end{pmatrix}$$

将第一行的 a_n 提出，然后除了第一行以外第 i 行加上第一行的 a_i 倍。

$$det(\mathbf{A}) = \begin{pmatrix} a_1a_n & a_2a_n & a_3a_n & \cdots & a_{n-1}a_n \\ 0 & a_2\sum\limits_i a_i & 0 & \cdots & 0 \\ 0 & 0 & a_3\sum\limits_i a_i & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1}\sum\limits_i a_i \end{pmatrix}$$

可以得到

$$det(\mathbf{A}) = \prod a_i \cdot \left(\sum a_i\right)^{n-2}$$

因此我们可以用 DP 来解决这个问题。 $f(i,j,k)$ 表示搞到树上 i 号点， i 的子树中已经选了 j 条边，与 i 连着的点数是 k ，选边的所有方案的 $\prod a_i$ 之和。转移可以依次枚举 i 的儿子来转移。假设儿子为 y ，则：

若 i 到 y 的边被选了，则 $f(i,j,k) \times f(y,u,v) \to f'(i,j+u+1,k+v)$

若 i 到 y 的边没被选，则 $f(i,j,k) \times f(y,u,v) \times v \to f'(i,j+u,k)$

可以将 f' 看成枚举 i 儿子时的滚动辅助数组。

转移的时候 j,k,u,v 的上限都是受到子树大小的限制的。假如使用了限制去枚举的话，复杂度是 $O(n^4)$

通过 f ，我们可以求出 g 。设 $h(i)$ 为有恰好给定树的 i 条边的数量。可以得出

$$h(i) = g(i) - \sum_{j>i} h(j) \times \binom{j}{i}$$

答案就是

$$\sum_{i \geq N-1-K} h_i$$

总复杂度为 $O(n^4)$