《树哈希》解题报告

杭州学军中学教育集团文渊中学 周康阳

题目描述

给定正整数 n, q, mod。保证 mod 是质数。

对于一棵以点 1 为根的有根树 T，设 $s(T)$ 为这棵树中最多能选出多少个互不同构的子树（也就是这颗树本质不同的子树个数），那么这个树的权值 $w(T) = q^{s(T)}$。

对于所有 $1 \le m \le n$，输出所有大小为 m，根为 1 的有标号树的权值之和对 mod 取模后的值。

两棵有根树 T_1、T_2 同构当且仅当他们的大小相等，且存在一个顶点排列 σ 使得在 T_1 中 i 是 j 的祖先当且仅当在 T_2 中 $\sigma(i)$ 是 $\sigma(j)$ 的祖先。

限制与约定

对于所有测试数据，保证 $n = 100, 10^8 \le mod \le 1.01 \times 10^9, 1 \le q < mod$，且 mod 是质数。

本题共 1 个子任务，每个子任务 100 分。你在每个子任务中的得分为该子任务所有测试点的得分的最小值。

你必须按照输出格式输出 n 个数，但是你可以输出错误的答案。如果你输出了 c 个正确的答案，那么你获得的分数按照如下方式计算：

- 对于 $0 \le c \le 20$，你的得分为 $2c$ 分；
- 对于 $21 \le c \le 60$，你的得分为 $c + 20$ 分。
- 对于 $61 \le c \le 100$，你的得分为 $\lfloor \frac{c}{2} \rfloor + 50$ 分。

时间限制 `4s`，空间限制 `2048MB`。

题解

本题是原创题。

算法 1

我会爆搜！时间复杂度 $\Theta(n^{n-1})$。

可以选择结合打表，根据实现可以做到 $n=9\sim 11$。

算法 2

我会更牛的爆搜！

爆搜所有不同构的有根树，并计算这种有多少这种有根树。

根据实现可以做到 $n=17\sim 20$。

算法 3

我会容斥！

我们按照 siz 从大到小的顺序，依次确定子树间本质相同的关系。

对于同一个 siz 的点，将其分成一些等价类，每个等价类内的点同构，等价类之间两两不同构。

然后进行集合划分容斥，钦定其中若干对等价类本质相同。

这样我们就又得到了一些新的等价类，每个等价类内的点本质相同，而等价类之间没有限制。

为了使每个等价类内的点本质相同，我们把同一个等价类内的点缩成一个。（注意这样做之后，一个点可能会有很多入边，而且等价类的子树也会被缩成一个）

完成了容斥的过程后，我们要计算连边的方案数。这个时候我们就要对 DAG 计数了。

（虽然题面里要求根为 1，但是我们可以先计算根任意的答案，然后给答案乘 $\frac{1}{n}$ 即可）

设 cnt_i 表示第 i 个点出现了多少次。那么我们要求对于除了根意外的所有点，连向第 i 个点的所有点的 cnt 之和是 cnt_i（可能有重边）。

因此可以爆搜 cnt 集合。设 f_i 为 $cnt=i$ 的点数，那么有 $\sum cnt_i=\sum if_i=n$，所以只会有 $\pi(n)$ 种这样的集合。

现在按照 cnt 从小到大的顺序，给每个点找入边。设现在大小为 i。

首先如果 $i=1$ 的点，这些点形成了一颗有根树的形状，因此对答案的贡献是 $q^{f_1}f_1^{f_1-1}$。

而对于 $i>1$，这个时候有两种点。一种点是只被一个 $cnt=i$ 的点连的点。另一种点是被一些 $cnt<i$ 的点连接的点。

第一类点的处理比较简单：这类点在分等价类时等价类大小是 1，因此对答案的贡献就是 q。而连边的方案，只要枚举二类点的数量 k，做一个 f_i 个点 k 个根的有标号有根树计数即可。

相对难算的是算 一个二类点做出的贡献。

这里我们一步一步做上面我们在开头做的事情。

首先先做第一步分组：设 $[x^t/t!]A(x)$ 为这个等价类有 t 个点的方案数。对于之前的每个等价类，都可以连 若干个当前等价类点 为儿子。那么可以列出 $A(x)=\prod_{p<i}(\sum_j \frac{x^{jp}}{(j!)^p})^{f_p}$。

带上 q 的贡献就是 $B(x)=q(A(x)-1)$。

然后我们进行了容斥：钦定若干个等价类是相同的。

一个比较经典的结论钦定成 t 个集合的容斥系数是 $(-1)^{t-1}(t-1)!$（容斥系数是所有有标号连通图的 -1 的边数次方之和。这个的 EGF 是 $\ln(1+x)$，提取 $[x^t/t!]$ 即可）。

此时 $G(x)=\sum_{i\geq 1}\frac{(-1)^i(i-1)!}{i!}B(x)^i=\ln(1+B(x))$。那么再提取 $[x^i]G(x)$ 就行了。

在做完上述过程后，由于集合之间是无序的，所以最后还要给答案乘以 $\prod_i \frac{1}{f_i!}$，以及给所有点分配标号的 $n!$。

时间复杂度 $\Theta(\pi(n)\operatorname{poly}(n))$，可以通过 $n=50\sim 70$。得分主要取决于复杂度中 $\operatorname{poly}(n)$ 的大小。

算法 4

我会 DP！

考虑对上面的过程做 DP。按照 cnt 从小到大加入元素。记录当前的向后面连了至少一条边的点的 cnt 集合 S 和当前点数 c。

在转移前，先容斥若干个元素，钦定他们没有出现；转移后，剔除掉不再向后面连边的点。

这样记录状态有个很大的好处就是 $\sum cnt_i\leq \frac{n}{2}$。用类似高维前缀和的方法转移，时间复杂度为 $\Theta(\pi(\frac{n}{2})\operatorname{poly}(n))$，但是 $\operatorname{poly}(n)$ 太大，因此不能得到更高的分。

在算法三的 $siz=1$ 那一层中，我们不进行集合划分容斥，因为这些点都是叶子。然后在 DP 中，我们只加入非叶子节点的那些 cnt。同时为了满足这些加入的点是非叶子，我们限制它们要么连了叶子要么往后面连了非叶节点。

这样每个 S 中的点向后连接的大小都 ≥ 2，因此时间复杂度就被优化至 $\Theta(\pi(\frac{n}{3})\operatorname{poly}(n))$。

我们进而考虑不容斥 $siz \leq B$ 的那些点。搜出所有 $siz \leq B$ 的有标号有根树，然后枚举哪些树出现了，然后容斥一下就变成限制只能出现一些树了。

我们 DP 中加入的那些 $siz > B$ 的点，我们还需要限制它们要么连的 $siz \leq B$ 的点的大小和 $\geq B$，要么连向了接下来在 S 中的新点。

这样时间复杂度为 $\Theta(\pi(\frac{n}{B+2})F(B)\operatorname{poly}(\mathrm{n})+H(B))$，其中 $F(B)$ 是 $siz \leq B$ 的子树的本质不同的选取状态数，前四项是 $1,2,6,24$。$H(B)$ 是搜索得到 $F(B)$ 的复杂度。标算选取了 $B=3$。

可以通过 $n=100$，期望得分 100 分。

总结

本题题面简洁，但是思维难度和代码难度都不低。覆盖算法面广，深刻地考察了组合数学、容斥原理、Prufer 序列、动态规划、搜索、自动机等知识点。同时使用了 SPJ 进行评分，有效地提升了本题的区分度。总而言之，这是一道不可多得的好题！