Tree Color 命题报告

杨天祺 李嘉图

1 题目

1.1 题目描述

给定一棵有标号有根树 T 和一个返祖边的集合 E。更严谨地，$E \subseteq T \times T$，并且对于任何 $(u, v) \in E$，都有 $u \neq v$，且 u 在 T 中是 v 的祖先。现在要将 T 上的每一条边染为黑色或白色。一个染色方案是合法的，如果对于任何 $(u, v) \in E$，在 u 到 v 的路径上至少存在一条边被染为了白色。求合法的染色方案数对某个给定的数取模的结果，$|T| \leq 3 \times 10^5$。

1.2 $O(n^2)$ 解法

定义关于 u 的部分染色方案为一个以 u 为根的子树内的边尚未确定染色、而其他边均已确定染色的方案；称一个关于 u 的部分染色方案是合法的，如果所有完全在 u 子树外的返祖边对应的路径上都存在一个白色边。定义一个部分染色方案的补全是对子树内边的染色；称补全是合法的，如果补全所得到的完整染色方案是合法的。

对于一个关于 u 的部分染色方案 $\mathcal{C}_u$，记 $p(\mathcal{C}_u)$ 为最小的 k，使得 u 的第 k 个祖先和第 $k+1$ 个祖先之间的边被染成白色。

定理 1.1. *令 $\mathcal{C}_u, \mathcal{C}'_u$ 是两个关于结点 u 的合法部分染色方案，且 $p(\mathcal{C}_u) = p(\mathcal{C}'_u)$，那么任何一个对于 $\mathcal{C}_u$ 合法的补全对于 $\mathcal{C}'_u$ 也是合法的。更进一步，对于它们的合法补全方案数是相等的。*

证明. 第二部分是第一部分的直接推论，仅证第一部分。如果对于 $\mathcal{C}_u$ 的补全是合法的，为证这一补全对于 $\mathcal{C}'_u$ 也是合法的，仅需考察三种返祖边对应的限制都被满足。

1. 完全在 u 为根之外的返祖边对应的限制：由于部分染色方案 $\mathcal{C}'_u$ 是合法的，这类边对应的限制已经被满足。
2. 在 u 为根的子树之内的返祖边对应的限制：由于在 u 为根的子树内的返祖边对应的限制是否被满足仅与子树内的染色方案有关，既然其在 $\mathcal{C}_u$ 的补全中被满足，那么在 $\mathcal{C}'_u$ 的补全中也被满足。
3. 跨过 u 的返祖边对应的限制：如果这一返祖边在 $\mathcal{C}_u$ 中已经被满足，由于 $p(\mathcal{C}_u) = p(\mathcal{C}'_u)$，其在 $\mathcal{C}'_u$ 中也已经被满足，因而在 $\mathcal{C}'_u$ 补全中已经被满足；如果其在 $\mathcal{C}_u$ 中未被满足，既然其在 $\mathcal{C}_u$ 的补全中被满足，其在子树内的部分在补全中已经有一条边染为了白色，因而也在 $\mathcal{C}'_u$ 的补全中被满足。

□

设 $dp[u][k]$ 表示关于结点 u 的、满足 $p(\mathcal{C}_u, u) = k$ 的部分染色方案 $\mathcal{C}_u$ 的合法补全方案数。根据以上定理，$dp[u][k]$ 是良定义的。如果存在一条 u 到其祖先且长度小于等于 k 的返

祖边，那么显而易见 $dp[u][k]=0$，因为这条返祖边对应的路径上没有任何白色边。否则我们容易说明

$$dp[u][k]=\prod_{v\in \mathrm{chl}(u)}(dp[v][k+1]+dp[v][0]),$$

其中 $\mathrm{chl}(u)$ 是结点 u 的儿子集合，对于叶节点自然有 $dp[u][k]=1$。这是由于 u 到其每一个儿子的边的染色方案是独立的，且对于其儿子 $v\in \mathrm{chl}(u)$，若其被染成白色，那么方案数是 $dp[u][0]$，否则为 $dp[u][k+1]$。最终答案即为 $dp[r][0]$，其中 r 为根节点。朴素实现这一做法的时间复杂度为 $O(n^2)$，因为每一条树上的边对应了一次 $O(n)$ 的计算。

1.3 $O(n\log n)$ 解法

分析动态规划状态间的依赖关系。注意到 $dp[u][k]$ 的计算依赖 $dp[v][k+1]$ 和 $dp[v][0]$，其中 $dp[v][0]$ 同 k 无关，并且 $(k+1)-\mathrm{depth}(v)=k-\mathrm{depth}(u)$。设 $f[u][k]=dp[u][k+\mathrm{depth}(u)]+dp[u][0]$，其中 $-n\le k\le n$，整理动态转移方程为

$$\begin{aligned}f[u][k]&=\prod_{v\in \mathrm{chl}(u)}f[v][k]+dp[u][0]\\&=\prod_{v\in \mathrm{chl}(u)}f[v][k]+\frac{1}{2}f[u][-\mathrm{depth}(u)].\end{aligned}$$

考虑下面的算法。对于每一个结点 u，维护一个值域在 $[-n,n]$ 的动态开点线段树（支持区间加、区间赋值、单点查询），其中第 k 位记录 $f[u][k]$。如果 u 的所有孩子对应的线段树均已完成计算，执行下面的算法。

1. 将所有线段树合并，得到一棵新树，其中第 k 位

$$T_u(k)=\prod_{v\in \mathrm{chl}(u)}f[v][k].$$

2. 根据状态转移方程，$f[u][-\mathrm{depth}(u)]=2T_u(k)=2dp[u][0]$，设 $d=T_u(k)$；
3. 设 u 到其祖先的最短返祖边长度为 l，区间赋值 $T_u(l-\mathrm{depth}(u)),\ldots,T_u(n)=0$；
4. 将整棵线段树加上 d。

而对于叶子，仅需用一次线段树区间赋值将 $T_u(l-\mathrm{depth}(u)),\ldots,T_u(n)$ 赋值为 1，其余部分赋值为 0 即可。注意到这里线段树仅作为承载标记的结构，并不需要求任何区间信息，因此可以在线段树合并的同时计算逐点乘积。注意到每合并一次所有线段树的结点个数减一，而每一次区间操作会增加 $O(\log n)$ 个结点，因此总的时间复杂度为 $O(n\log n)$。