

D 拨打电话

陈鸿基 / SpiritualKhorosho

Tsinghua University

May 26, 2024

简要题意

定义一棵有根树上从叶结点 u 向叶结点 v 拨打电话的正确号码是沿着树上最短路径依次连接相应字符串的结果. 给定一棵有 n 个结点的有根树和 q 组询问, 每组询问为从一个结点拨打指定的电话号码, 可以匹配上多少结点.

$2 \leq n \leq 10^5, 1 \leq q \leq 10^5$, 树上串及询问串总长分别不超过 3×10^6 .

观察

由于需要比较的号码可能很长, 不难想到用 Hash 来加速字符串比较的过程. 即, 如果我们能快速求出从一个叶结点出发, 到其它所有叶节点的路径对应的号码的 Hash 值, 我们就可以通过这个 Hash 值来判断有多少叶结点匹配询问的电话号码.

观察

由于需要比较的号码可能很长, 不难想到用 Hash 来加速字符串比较的过程. 即, 如果我们能快速求出从一个叶结点出发, 到其它所有叶节点的路径对应的号码的 Hash 值, 我们就可以通过这个 Hash 值来判断有多少叶结点匹配询问的电话号码.

直接维护本题给出的号码比较困难, 但我们仍然可以考虑使用点分治的方法来处理路径问题.

把前缀 b_i, c_i, d_i 都看成是对 Hash 值的一个变换, 那么一条路径对应的号码相当于这些变换的复合. 如果我们使用一些可逆的变换 (如最常用的线性函数 $f(c, hash) = w(c) + a \cdot hash$), 我们就可以对询问串 (的 Hash 值) 逆变换得到: 如果一个电话经过当前重心, 则从重心往后, 仍需处理的电话号码 (的 Hash 值). 这样, 我们就可以在重心处比较去掉部分前缀的查询串和加上部分前缀的电话号码是否相等.

设计 Hash

如果只有 b_i 和 c_i , 则相应的 Hash 函数可以直接看成单向边的边权. 对于各叶结点的电话号码 a_i , 从 f_i 到 i , 需要使用 c_i 进行变换, 而从 i 到 f_i 则需使用 b_i 变换. 对于询问串, 则应分别用相应的逆进行变换.

加上 d_i , 问题变得稍微更复杂一些, 但仍然可以在原来的基础上实现. 注意到只有从 f_i 到 i 时不需要使用 d_i , 我们对指向子结点的边的 c_j 变换左复合 d_i , 而对 f_i 到 i 的 c_i 右复合 d_i^{-1} 即可. 这样只有在子树内会计入 d_i 的贡献, 而从父结点进入子树时贡献可以被抵消掉.

总复杂度

$$O((n + q) \log n + \sum |a_i| + \sum |b_i| + \sum |c_i| + \sum |d_i| + \sum |r_i|).$$