

Substrings on a Tree 解题报告

绍兴市第一中学 孙耀峰

1 试题来源

Codechef April 2012 TSUBSTR

2 试题大意

给出一棵 N 个节点的树，每个节点上有一个字母。定义树的子串为树上某一对点 u, v ，从 u 点到 v 点路径上点的字母连接而成的字符串。其中 u 点是 v 点树上的祖先。

题目要求回答该树有多少不同的子串，并将进行 Q 次询问。每次给出一个排列 P ，表示字母 ' a ' - ' z ' 的大小关系。要求求出将该树所有不同的子串按照这个大小关系的字典序排序后的第 K 小子串。

保证最后的输出 $Output \leq 800KB$ 。

数据范围： $N \leq 250000, Q \leq 50000, K \leq 2^{63} - 1$ 。

3 算法介绍

3.1 问题简化

我们先考虑这个问题的一個简化版本。

假设题目给出的是一条链而不是一棵树。这样问题就变成了询问一个字符串有多少不同的子串和给定大小关系，询问第 K 小的子串。

对于这两类询问，都是后缀自动机 (SAM) 算法中非常经典的问题。大致是这样的流程：

对字符串建立 SAM ，根据 SAM 中的 $Parent$ 数组建立 $Parent$ 树。对于第一类询问，可以通过求解 $Right$ 数组，并在 $Parent$ 树上进行简单的树形 Dp 来统计。对于第二类操作，可以利用 SAM 中的 $Trans$ 数组对 SAM 进行 DFS 来解决。

解决的方法非常多，此次不再赘述，详情可以参见 2014 年的集训队论文中关于 SAM 的介绍。

3.2 算法思路

现在考虑原问题，即一棵树的情况。

对于树的情况，非常容易联想到广义的后缀自动机。这其实就是对字典树建立后缀自动机。与序列问题不同的是，树上问题我们需要对于树上每一个节点记录一下这个节点在 SAM 中所对应的点。之后的插入并不能像序列问题那样直接插入在最末端，而应该从它树上父亲在 SAM 中所对应的点处插入。新建节点的信息可以直接覆盖之前节点创建的信息。

通过这种方法就可以得到字典树的后缀自动机。至于题目要求的两类询问，直接效仿之前的做法即可。

时间复杂度 $O((N + Output) * 26)$ ，空间复杂度 $O(26 * N)$ 。