

# Codechef DEC12 Different Trips 解题报告

镇海中学 邹逍遥

## 1 试题来源

Codechef DEC12 DIFTRIP

## 2 问题简述

给定一棵根含有 $n$ 个点的根为1号点的树，每个结点上有一个数字，这个数字等于这个结点的度数。

现在需要选出尽可能多的路径满足：

- 任意一条路径都是从根到某个点的路径的一部分。
- 任意两条路径上标注的数字按顺序排列后形成的字符串不同（长度不同或任意一个位置不同）。

## 3 数据范围

$$1 < N < 10^5$$

## 4 解答

假如这道题的树形态是一条链，那么就是简单的子串数量统计问题，可以用后缀数据结构轻松解决。那么问题的关键就变成了如何将序列数据结构扩展到树上。

### 4.1 使用后缀自动机解决

注意到这些串都是从根到某一个点的串的子串，那么只需要在树上建立出后缀自动机即可查询子串数量。

建立方法与对字符串建立后缀自动机类似。在字符串上建立时我们会对于每个前缀记录下代表这个串的那个节点，然后在插入下一个字符时在这个节点的基础上进行添加。在树上建立时同样记录下代表每个“前缀”（即根到一个点的一条路径）的节点，插入新点时在它的父亲的节点后插入即可。

在后缀自动机中查询子串数量是一个很基本的操作，只需要将每个点代表的子串数量相加（这个点的length减去它fail的length）。

但是由于单个字符最大可能到 $O(n)$ 级别，使用离散化后字符集大小变为不同字符数量。容易看出这是 $O(\sqrt{n})$ 的。这样做的时间复杂度为 $O(n)$ 空间复杂度为 $O(n\sqrt{n})$ 。使用map存储可以以增大时间消耗来减小空间消耗，做到时空复杂度均为 $O(n \log n)$ 。

虽然相比后缀数组解法空间复杂度较大但是代码实现十分简单。

#### 算法效率

时间复杂度：  $O(n \log n)$

空间复杂度：  $O(n \log n)$

## 4.2 使用后缀数组解决

当然本题也可以不使用后缀自动机，而使用后缀树组解决。

在树上建立后缀数组仍然可以使用倍增法。具体做法与序列上类似，只是在倍增时将某个后缀的向后 $2^i$ 位置代表的后缀替换成它的 $2^i$ 的祖先代表的串。这样就可以将每个点到根的字符串进行排序。height数组的求法也和序列上类似，只是条件变为了一个串的height不大于它的父亲节点代表的串的height+1。

注意到这样做其实是求出了“叶子节点个数”的串的后缀数组的合并。对于这个“后缀数组”，仍然可以使用正常的统计方法统计出这些串包含的不同字符串数量。

设 $n$ 个字符串为 $s_i$ ，则答案为 $\sum \text{length}(s_i) - \text{height}(s_i)$ 。

### 算法效率

时间复杂度： $O(n \log n)$

空间复杂度： $O(n)$