

Stringology is Magic解题报告

宁波市镇海中学 邹逍遥

1 试题来源

Codechef October Challenge 2014

<http://www.codechef.com/OCT14/problems/QSTRING>

2 试题大意

给出一个长为 n 的小写字母串 T 和 m 个询问，询问有两种：

- 给定 k_1, k_2 ，询问在所有 T 的子串中，字典序排在第 k_1 位（去重后）的串，假如有多个输出第 k_2 小的一个。
- 给定 l, r ，询问子串 $T[l \cdots r]$ 字典序在 T 的所有子串中的排名 k_1 ，以及在相同的串中的排名 k_2 。

相同的串按在 T 中的出现顺序排列。

3 数据范围

$$1 \leq n \leq 10^6$$

$$1 \leq m \leq 10^6$$

4 算法介绍

4.1 将所有子串排序

考虑处理出 T 的后缀数组 SA 和 $height$ 数组 h ，那么 T 的不同子串个数就是总子串个数 $-\sum h_i$ 。

同理可以求出后缀数组中的前缀代表的所有子串中有多少个不同子串。因为相邻两个串中的重复串数量在`height`数组中已经求出。

所以可以预处理出不同子串数的前缀和，这样就可以在前缀和中二分来找到第 k_1 大的串是什么。

4.2 解决第一问

预处理出这些东西以后我们能知道第 k_1 大的串是什么，并且知道它第一次出现的位置，那么通过二分+RMQ我们可以知道在后缀数组中这个串最后一个出现的位置。

那么答案就是后缀数组这个区间内子串左端点第 k 大的答案。

而这道题是不需要修改的，也就是说只需要支持静态查询区间第 k 大即可。这个可以用一个可持久化线段树轻松解决。

这样第一问就在 $O(n \log n + m \log n)$ 的时间内解决了。

4.3 解决第二问

考虑怎么实现第二问：首先需要找出这个串在后缀数组中出现的位置，那么通过在后缀数组中二分查询RMQ就能找到这个串第一次出现的位置，通过预处理的不同串数量的前缀和可以很方便地算出 k_1 。

那么考虑怎么算出 k_2 ：首先同理可以二分求出这个串最后一次出现的位置，那么只需要求出在这一段中左端点比询问的左端点小的有几个即可。

这个询问也可以通过可持久化线段树解决。

那么这道题就在 $O(n \log n + m \log n)$ 的时间内解决了。