

To Queue or not to Queue 题解

雅礼中学 袁宇韬

2015 年 11 月 3 日

1 题目大意

给定一个初始为空的字符串，要求支持在最后添加字符和删除第一个字符，每次操作后询问本质不同的子串个数。操作数 ≤ 1000000 。

2 题解

如果只要求出一个字符串的本质不同的子串，可以使用后缀数组、后缀树、后缀自动机等算法轻松解决。由于本题需要支持在最后添加字符和删除第一个字符，用后缀自动机难以解决，考虑使用其它算法。

如果构造出字符串的后缀树，则本质不同的子串个数为后缀树的所有边的长度之和。使用后缀树的在线构造算法（Ukkonen 算法），可以在最后添加字符时维护后缀树。Ukkonen 算法的相关资料可以在网上查到，这里不再赘述。

现在考虑删除第一个字符的操作。这个操作相当于在后缀树中删除最长的后缀。这个后缀一定对应后缀树中的一个叶子结点。在后缀树中删除一个后缀相当于在 Trie 树中删除一个字符串。只需要删除这个串对应的结点，然后如果之前这个结点的父亲结点变为叶子结点则需要继续用同样方式删除，直到父亲结点不是叶子结点为止。即需要删除所有只属于这个串的结点。

注意到在 Ukkonen 算法中我们需要维护最长的没有对应叶子结点的后缀在树上的位置。如果这个串所在的边被删除，则这个后缀会对应一个新的叶子结点。此时应该新建一个叶子结点，并用类似添加字符时的方法建立后缀链。同时由于这个结点属于另外一个串，应该停止删除结点。

这样可以在线解决这个问题，时间复杂度为 $O(q)$ 。

本题也可以用后缀数组解决。首先离线得到所有插入的字符，然后对这个字符串倒序求出后缀数组。这样插入字符相当于添加一个更长的后缀，删除字符相当于使所有后缀长度减 1。一个字符串的本质不同的子串个数为所有后缀的长度之和减去按照字典序排序后所有相邻后缀的最长公共前缀（LCP）之和。

在插入后缀时可以找到后缀数组中的前一个已经插入的后缀，更新这个后缀与前后两个已经插入的后缀的 LCP。删除字符时所有后缀的长度会减少 1，并且所有相邻两个后缀满足一个是另一个的前缀时 LCP 会减少 1。可以在有相邻后缀满足一个是另一个的前缀时删除较短的后缀，因为这个

后缀不会贡献本质不同的子串。这些操作可以用线段树维护，LCP 可以在 height 数组上求 RMQ 得到。这个做法的时间复杂度为 $O(q \log q)$ 。我没有实现这个算法，但是如果常数较小应该可以通过本题。

3 官方题解

见<https://discuss.codechef.com/questions/23739/tmp01-editorial>