发言稿

各位老师，各位同学：

大家晚上好。我是罗穗骞，来自广东华南师大附中。今天我给大家介绍一种处理字符串的有力工具，（点）后缀数组。

我的论文分两部分，第一部分介绍一种后缀数组的构造方法：DC3 算法。第二部分通过分析一道例题，介绍后缀数组的具体应用。

我们先来看一下关于后缀数组的基本定义。后缀数组保存的是一个字符串的所有后缀的排序结果，名次数组保存的是所有后缀的名次。

下面以字符串“aabaaaab”为例。

先写出这个字符串所有的后缀，（点）然后按照字典序对它们进行排序。

排序后将结果放在 sa 数组中。（点） Sa[1]=4 说明排名第 1 的字符串是以 4 个字符开始的后缀。rank[2]=6 说明以第 2 个字符开始的后缀排名第 6。简单的说，后缀数组是“排第几的是谁？”，名次数组是“你排第几？”。容易看出，后缀数组和名次数组为互逆运算，名次数组可用于字符串大小比较。

接下来介绍一种后缀数组的构造方法——DC3 算法。（点）可能有的同学认为这个方法复杂，即使理解了也难以实现。（点）针对这个问题，我在论文中除了介绍算法，还重点介绍这个算法的具体实现，（点）并同时给出了一个仅 40 行的实现代码。

我们先看看 DC3 算法的主要步骤。将后缀分成两部分，先对第一部分的后缀排序，再对第二部分的后缀排序，然后合并得到最终结果。整个思路和归并排序有些类似。（点）我们先看第（1）步。

将后缀按起始位置模 3 的余数进行划分。为了操作方便，这里假设字符的编号从 0 开始。（点）以这个字符串为例，绿色字符开始的后缀为第一部分，白色字符开始的后缀为第二部分。

接下来将要对绿色字符开始的后缀进行排序，这里要求字符串必须以一个最小的字符结尾。

做法是将后缀 1 和后缀 2 连接，（点）如果这两个后缀的长度不是 3 的倍数，那先各自在末尾添 0 使得长度都变成 3 的倍数。然后每 3 个字符为一组，（点）用基数排序将每组字符“合并”成一个新的字符。（点）然后用递归的方法求这个新的字符串的后缀数组。

在求出新的字符串的后缀数组之后，（点）便求出了这些绿色字符开始的后缀，（点）也就是求出了原字符串所有起始位置模 3 不等于 0 的后缀的排序结果。（点）步骤（1）完成。

接下来看第（2）步。（点）

对所有起始位置模 3 等于 0 的后缀进行排序。（点）每个后缀都可以看成是（点）一个字符和一个绿色字符开始的后缀，这个字符作为第一关键字（点），这个后缀在第（1）步中的 rank 值作为第二关键字（点），一次基数排序即可以完成对所有起始位置模 3 等于 0 的后缀的排序，（点）步骤（2）完成。

然后看第（3）步。（点）

合并前两步的排序结果。这个过程和归并排序的合并操作类似。后缀的大小比较有两种情况，（点）第一种情况是起始位置模 3 等于 0 的后缀和起始位置模3等于 1 的后缀比较，先比较第一个字符（点），再比较接下来的后缀（点）。第二

种情况是起始位置模 3 等于 0 的后缀和起始位置模 3 等于 2 的后缀比较（点），先比较前两个字符（点），再比较接下来的后缀（点）。无论哪种情况，（点）每次的比较都可以高效的完成 步骤（3）完成。

下面对这个算法的时间复杂度进行分析。假设这个算法的时间复杂度为 f(n)。容易看出，除了递归处理的那一步，其余的时间均为 O(n)。递归处理时，新的字符串的长度不会超过 2n/3，那么有：

f(n)=O(n)+f(2n/3)，最后计算得到 f(n)=O(n)。所以 DC3 算法是一个优秀的线性算法。

接下来通过一道题目，一起来看看后缀数组在具体问题中是如何运用的。题目是求两个字符串的最长公共子串。在介绍做法前，先介绍后缀数组的一些性质。

定义 height 数组表示排名相邻的两个后缀的最长公共前缀。那么 height 数组有以下性质：

两个后缀的最长公共前缀为排名在它们之间的后缀的 height 值中的最小值。

如图所示，如果这两个字符串的最长公共前缀是 1，首先 height 值中的最小值不会小于 1，（点）因为它们的第一个字符是相同的（点）。其次 height 值中的最小值也不会大于 1，如果至少是 2，那么说明它们至少前两个字符都是相同的，（点）这与最长公共前缀是 1 相矛盾（点）。所以求最长公共前缀问题便转化为了求区间最小值问题。

回到例 1，求 A 和 B 的最长公共子串等价于求 A 的后缀和 B 的后缀的最长公共前缀的最大值。把这两个字符串连起来，看看能不能从后缀数组中找到规律。

如图所示。那么是不是 height 数组中的任意一个值都有可能成为答案呢？（点）不是的（点），还必须满足一个条件，那就是这排名相邻的两个后缀要在不同的字符串中（点）。所以答案应该是（点）排名相邻的且不在同一个字符串中的 height 值中的最大值。（点）

还有一个问题没有解决，如何高效的求出 height 数组？定义 h 数组，h[i]表示后缀 i 和排在它前一名的后缀的最长公共前缀。那么 h 数组有以下性质：

h[i]≥h[i-1]-1。证明如下，（点）假设排在后缀 i-1 前一名的是后缀 k（点）最长公共前缀为 h[i-1]，现在考虑后缀 i，（点）后缀 k+1 还一定在它的前面（点），它们的最长公共前缀为 h[i-1]-1，但是它们之间也可能还有别的后缀（点），无论有没有，后缀 i 和它前一名的最长公共前缀至少为 h[i-1]-1（点），所以原不等式得证。

利用这个性质，height 数组便可以在 O(n)的时间内求出。

再回到例 1，求后缀数组，求 height 数组，求最大值的时间复杂度都是原字符串的长度和。（点）时间复杂度已经取到下限，这是一个非常优秀的做法。

后缀数组是一种处理字符串的有力工具。我们应该掌握好后缀数组这种数据结构，并且能在不同类型的题目中灵活、高效的运用。

在我论文里，你能看到的内容有：两种构造后缀数组的算法，和它们完整的代码（点），其中倍增算法为 25 行，DC3 算法为 40 行（点）。还有后缀数组在不同类型的题目中的应用。一共有 13 道题（点），每题都有原题，解答和参考程序（点）。欢迎阅读。

我的论文介绍完毕，谢谢大家。