后缀数组

参考 <http://www.nocow.cn/index.php/%E5%90%8E%E7%BC%80%E6%95%B0%E7%BB%84>

<http://www.cnblogs.com/mycapple/archive/2012/08/09/2630424.html>

1. 字符串aabaaaab的所有后缀

[](http://www.nocow.cn/images/a/a2/20120630fc_suffix.png)

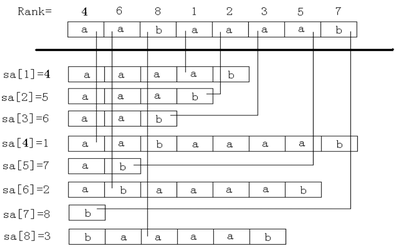
1. 后缀数组基本包含两个数组 数组SA 和数组Rank

后缀数组SA 是一个一维数组，它保存1..n 的某个排列SA[1]，SA[2]，……，SA[n]，并且保证Suffix(SA[i]) < Suffix(SA[i+1])，1≤i<n。也就是将S 的n 个后缀从小到大进行排序之后把排好序的后缀的开头位置顺次放入SA 中。其中，suffix(i)表示字符串s[i,i+1…n-1]，即字符串s起始于第i个字符的后缀。

名次数组Rank[i]保存的是Suffix(i)在所有后缀中从小到大排列的“名次”。

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。

Sample：



**使用倍增算法来计算Rank 然后通过Rank计算Sa**

## 倍增算法

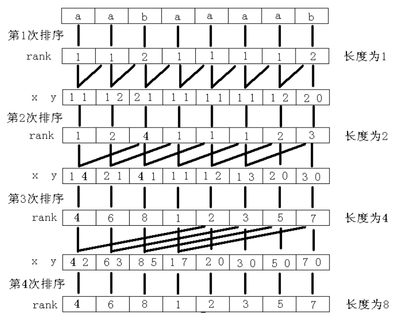
一、主要思路：[倍增](http://www.nocow.cn/index.php?title=%E5%80%8D%E5%A2%9E&action=edit&redlink=1)，s[*i*..*i* + 2*k* − 1]的排名通过s[*i*..*i* + 2*k*− 1 − 1]和s[*i* + 2*k*− 1..*i* + 2*k* − 1]的排名得到。

二、简要过程：已知每个长度为2*k*− 1的字符串的排名，则可作为每个长度为2*k*的字符串求排名的关键字xy，s[*i*..*i* + 2*k* − 1]第一关键字x为s[*i*..*i* + 2*k*− 1 − 1]的排名，第二关键字y为s[*i* + 2*k*− 1..*i* + 2*k* − 1]的排名。以字符串aabaaaab为例：

1. k=0,对每个字符开始的长度为20 = 1的子串进行排序，得到rank[1..8]={1,1,2,1,1,1,1,2}
2. k=1,对每个字符开始的长度为21 = 2的子串进行排序：由k=0的rank得关键字xy[1..8]={11,12,21,11,11,11,12,20}，得到rank[1..8]={1,2,4,1,1,1,2,3}
3. k=2,对每个字符开始的长度为22 = 4的子串进行排序：由k=1的rank得关键字xy[1..8]={14,21,41,11,12,13,20,30}，得到rank[1..8]={4,6,8,1,2,3,5,7}
4. k=3,对每个字符开始的长度为23 = 8的子串进行排序：由k=2的rank得关键字xy[1..8]={42,63,85,17,20,30,50,70}，得到rank[1..8]={4,6,8,1,2,3,5,7}

**注意：在排序过程中，rank[]可以有相同排名，但是sa[]排第几是没有相同的（就像Excel的排序，sa相当于编号，rank相当于排名）。这点可以从程序中体现。建议读者跟踪一下程序体会一下。**

**整个过程**如图：



倍增算法思想

字符串字典序排序

**体会下最终结果 SA数组是不会有重复的排名的**