**[字符串匹配 后缀数组 trie图(更新)](http://www.cppblog.com/wolf5x/archive/2009/07/16/90269.html)**

字符串少量习题小结.  
  
spoj694(易) 后缀数组  
求一个字串的不同子串个数.  
按rank考虑子串.加入子串S[i]时,获得了len-Sa[i]个不同子串.但其中height[i]个已经属于S[i-1]了,所以实际子串数增加了len-Sa[i]-S[i-1]. 顺序扫一遍height数组即得解.  
  
spoj687(中) 后缀数组  
求一个串的重复次数最多的连续重复子串.  
设周期为L的连续重复子串存在,则点0,L,2L,...,kL必能覆盖到一个完整周期. 因此对L,考察这些点的字符相等情况,LCP情况,可得到L的解.  
枚举L.  
复杂度是O(n/1+n/2+...+n/n) = O(nlogn)  
  
pku3693(中) 后缀数组  
同spoj687,只是结果还要输出字典序最小的满足条件的串.可以借助rank数组直接比较字典序.只是要注意在考察点kL时,要把以(k-1)L+1,...,(k+1)L-1为起点的子串都访问一遍找最小rank者.  
  
pku1743(中) 后缀数组  
找一个串的最长不重叠相同子串.  
由于某子串可能整体加上(或减去)相同偏移量,因此不直接对原串操作,而是构造新串b, 其中b[i]=a[i]-a[i-1]. 此时求得最长不重叠相同子串的长度+1便是结果.  
可以二分长度,或者栈扫描(\*)直接求最大长度.  
  
whu1084(易) 后缀数组  
求重复次数最多的不重叠子串长度  
spoj687的简单版,不要求循环节连续,直接二分长度即可.  
  
pku2778(易) 多串匹配+DP AC自动机,trie图  
字符集大小为4, 给出m个(m<=10)禁止单词(长度<=10), 求长度为n(n<=2\*10^9)的不包含任何禁止单词的串的个数.  
对禁止单词建立trie图,并计算出图中任意合法结点之间的转移数,这样便求得1步转移矩阵.  
做n次方后的矩阵,第1行中属于合法状态的元素之和即为解.  
禁止单词总长度不超过100,因此合法状态亦<100.总复杂度100^3\*logN  
  
zju3228(中) Searching the String 后缀数组,AC自动机,trie图  
原串长10^5, 现在有10^5次查询, 每次查询一个长度<=6的模式串在原串中的最大匹配次数.  
模式串的匹配方式有可重叠和不可重叠两种, 需针对查询的类型返回相应值.  
后缀数组解法(在线):  
对原串建立sa和height数组.由于模式串长度最大只有6, 我们可以将height数组分别按L=1..6分组,预处理求出相应长度每组内不重叠子串的最大匹配次数,此过程O(6\*nlogn).  
另外由于sa数组将所有后缀按字典序排好了,所以对一个询问, 可以二分找到它在sa中第一次出现的位置p1和最后一次出现的位置p2, 则p2-p1+1就是可重叠匹配的答案. 对不可重叠匹配,只需直接返回p1处预处理时的值. 每次查询O(logn).  
trie图,AC自动机解法(离线):  
把所有查询建trie图, 对图中的每个有效结点维护:该串长度,两类查询的计数,该串上一次被匹配的位置, 还要用个链表记下这个串属于哪些查询.  
剩下的就是经典的自动机多串匹配了.  
  
  
(\*)关于栈扫:  
height数组具有区间性,各个不同前缀被相应的极小值隔开,而一个区间中又有多个子区间.各区间值大于区间端点的部分互不影响.因此可以维护一个存放height值不减的栈,栈中每个元素的附属值, 记录了它在栈中相邻的两个元素为端点的连续区间内所有height值不小于它的必要信息.比如此题要记录height>=k的连续区间内sa[i] 的最大值和最小值.  
栈扫描的经典例子移步pku2559.  
  
  
(\*\*)trie图备忘:  
比trie树多了个后缀指针psuf. 设当前结点字母为c, 则psuf指向父亲的后缀的pch[c].  
trie树中的后代结点指针pch(已经更名为状态转移指针),当相应后代存在时,指向后代;否则指向当前结点的后缀的相应后代,即pch[k]=node[pa].pch[k].  
后缀指针: 在接下来的状态转移中,当前结点与它的后缀结点等价.  
后代结点指针: 在当前状态下,接收到字符ch时,转移到pch[ch]指向的结点.