**zz 后缀数组**

[acm](http://blog.163.com/lfw2565295@126/blog/#m=0&t=1&c=fks_080065086084080075092074087065086086081070085080) 2009-08-09 14:05:53 阅读587 评论0   字号：大中小 订阅

ACM/ICPC   2009-08-05 22:51   阅读10   评论1   字号： 大大  中中  小小 单独把它列出来是因为这个东西真的很神奇~~~

后缀数组经典思想：多串合并+二分答案+最优性--->可行性

例 1 ：最长公共前缀  
给定一个字符串，询问某两个后缀的最长公共前缀。   //  直接套用，ans=min( height[ i ] )+rmq    k<i<=j

例 2 ：可重叠最长重复子串  
给定一个字符串，求最长重复子串，这两个子串可以重叠。   //  ans=max( hegiht[ i ] )  0<=i<len

例 3 ：不可重叠最长重复子串（ pku1743 ）      AC  
给定一个字符串，求最长重复子串，这两个子串不能重叠。   // 二分转化为判定性问题

例 4 ：可重叠的 k 次最长重复子串（ pku3261 ）       AC  
给定一个字符串，求至少出现 k 次的最长重复子串，这 k 个子串可以重叠。   //  同上，也是二分

例 5 ：不相同的子串的个数（ spoj694,spoj705 ）  
给定一个字符串，求不相同的子串的个数。      
[解法]：  
对于每一次新加进来的后缀 suffix(sa[k]), 它将产生 n-sa[k]+1 个新的前缀。但是其中有  
height[k] 个是和前面的字符串的前缀是相同的。所以 suffix(sa[k]) 将 “ 贡献 ”  
出 n-sa[k]+1- height[k] 个不同的子串。累加后便是原问题的答案。这个做法  
的时间复杂度为 O(n) 。

例 6 ：最长回文子串（ ural1297 ）  
给定一个字符串，求最长回文子串。  
[解法]：  
将整个字 符串反过来写在原字符串后面，中间用一个特殊的字符隔开。这样就把问题变为 了  
求这个新的字符串的某两个后缀的最长公共前缀。  
eg：aabebf  ---->   aabebf&fbebaa

例 7 ：连续重复子串 (pku2406)             AC  
给定一个字符串 L ，已知这个字符串是由某个字符串 S 重复 R 次而得到的，  
求 R 的最大值。  
[解法]：  
做法比较简单，穷举字符串 S 的长度 k ，然后判断是否满足。判断的时候，  
先看字符串 L 的长度能否被 k 整除，再看 suffix(1) 和 suffix(k+1) 的最长公共  
前缀是否等于 n-k 。  
hit：此题更好的是考察KMP的next  
            int k=len-next[len];  
            if(len%k==0) fprintf(fout,"%d\n",len/k);  
            else fprintf(fout,"1\n");

例 8 ：重复次数最多的连续重复子串 (spoj687,pku3693)   //  还未完成，抽时间好好做，先做SPOJ  
给定一个字符串，求重复次数最多的连续重复子串。  
[解法]：  
先穷举长度 L ，然后求长度为 L 的子串最多能连续出现几次。首先连续出 现  
1 次是肯定可以的，所以这里只考虑至少 2 次的情况。假设在原字符串中连续 出  
现 2 次，记这个子字符串为 S ，那么 S 肯定包括了字符 r[0], r[L], r[L\*2],  
r[L\*3], …… 中的某相邻的两个。所以只须看字符 r[L\*i] 和 r[L\*(i+1)] 往前和  
往后各能匹配到多远，记这个总长度为 K ，那么这里连续出现了 K/L+1 次。最 后  
看最大值是多少。

例 9 ：最长公共子串 (pku2774,ural1517)     // 发现倍增的O(NlogN) 好慢……2500+ms  
给定两个字符串 A 和 B ，求最长公共子串。  
连接字符串，O(N)扫描

例 10: 长度不小于 k 的公共子串的个数 (pku3415)     //  未解决  
给定两个字符串 A 和 B ，求长度不小于 k 的公共子串的个数（可以相同）。  
<http://hi.baidu.com/zfy0701/blog/item/f2278a0928991dca3bc763a0.html>

例 11: 不小于 k 个字符串中的最长子串 (pku3294)  
给定 n 个字符串，求出现在不小于 k 个字符串中的最长子串。    //  已经AC，等于没AC，越来越觉得sort慢了……压着时限过的……且写了5K  
[解法]：  
参见例3：扩展到看k个一样  
   
例 12: 每个字符串至少出现两次且不重叠的最长子串 (spoj220)    //  未做，先去学了radix sort再考虑吧，不过SPOJ时空一向很宽……，那次16数独用了100MB……2s  
给定 n 个字符串，求在每个字符串中至少出现两次且不重叠的最长子串。  
[解法]：  
做法和上题大同小异，也是先将 n 个字符串连起来，中间用不相同的且没 有  
出现在字符串中的字符隔开，求后缀数组。然后二分答案，再将后缀分组。判 断  
的时候，要看是否有一组后缀在每个原来的字符串中至少出现两次，并且在每 个  
原来的字符串中，后缀的起始位置的最大值与最小值之差是否不小于当前答案  
（判断能否做到不重叠，如果题目中没有不重叠的要求，那么不用做此判断）。  
这个做法的时间复杂度为 O(nlogn) 。

例 13: 出现或反转后出现在每个字符串中的最长子串 (PKU1226)   //  AC  数据极弱，比我的算法弱  
给定 n 个字符串，求出现或反转后出现在每个字符串中的最长子串。  
[解法]：  
连接字符串（正反向），二分答案，判断是否都出现过。

后缀数组题目推荐  
1412  
后缀数组的题目

Uva11475

题目大意 给定一个字符串在字符串的末尾加最少的字符是的该字符串成为回文串。  
[这个题目我已经加到OJ上了 题号是1139 这个题目根本不用后缀数组的 而且后缀数组的效率也是不最高的 ——Sempr补充]

Pku2774 : 求两个字符串的最长公共子串长度。                //  AC

Whu1069  求两个字符串的最长公共子串长度。

pku3581 ：<http://acm.pku.edu.cn/JudgeOnline/problem?id=3581>      
题目大意：给定一个数组{A1, A2, …, An} 满足A1 > A2, …, An,把该数组分成三段，单独翻转，使得数组的字典序最小。

        
<http://acm.pku.edu.cn/JudgeOnline/problem?id=3623>              //  AC

题目大意：给定一个字符数组，可以从数组的开头或者结尾取出元素，按取出顺序排成一列，使得他的字典序最小。

<http://acm.pku.edu.cn/JudgeOnline/problem?id=1743>    //  AC 为什么G++挂，C++AC?肯定是我写错了……而且跑得很慢……  
题目大意：求最长不重叠差为定值子串的长度

-》最长不重叠重复子串的长度

(1)二分答案

（2）线性扫描

<http://acm.pku.edu.cn/JudgeOnline/problem?id=3450>      //    未作

题目大意：求多个字符串的最长公共字串。

<http://acm.pku.edu.cn/JudgeOnline/problem?id=3080>        //  AC

题目大意：求多个字符串的最长公共子串（弱）。

用后缀数组比较麻烦，可以枚举答案，用kmp判定。

Waterloo：life forms：<http://acm.pku.edu.cn/JudgeOnline/problem?id=3294>     //   AC  好题

题目大意：超过一半的串的公共最长子串。

<http://acm.pku.edu.cn/JudgeOnline/problem?id=3415>            //  未作   好题

题目大意：求两个字符串的长度大于k公共字串的个数。

Whu1084 <http://acm.whu.edu.cn/oak/problem/problem.jsp?problem_id=1084>

题目大意：求最长不重叠重复字串的长度和个数。

Toj2171 <http://acm.tju.edu.cn/toj/showp2171.html>

题目大意：统计每子串可分成可分成多少个重复串

<http://acm.pku.edu.cn/JudgeOnline/problem?id=2758>  
题目大意：给定一个串，要求支持两种操作：插入单个字符或者询问从两个位置开始的最大匹配长度。