字符串

YAZID

常见算法/数据结构

KMP

Hash

Trie树

AC自动机

manacher

后缀数组

后缀自动机

*回文树/回文自动机/后缀平衡树

KMP

单模匹配算法（单个模式串）

用p[i]表示在位置i匹配失败时，应跳转到哪个位置继续匹配。

线性预处理p。

线性匹配。

HASH

OI中最常用的字符串哈希算法：RK-Hash

把字符串视为一个base进制的大整数，求其模P意义下的值（P一般取质数）

即sum[i]=(sum[i-1]*base+st[i]) mod P

基本功能：O(1)提取一段子串的hash值/O(1)合并两个串的hash值

单哈希被卡的可能：

1. 自然溢出
2. 生日攻击
3. hack赛制

log时间求LCP

LCP=Longest Common Prefix(?)

二分+hash

例1 火星人prefix（bzoj1014）

给定一个字符串，要求支持3种操作：

1. 求两个后缀的最长公共前缀
2. 修改一个位置
3. 在两个字符中间插入一个新字符

Q<=150000

字符串长度始终不超过10^5

例1 火星人prefix

平衡树维护哈希

最长公共前缀？二分！

复杂度nlog^2

Trie树&AC自动机

Trie树（字典树）→AC自动机

（BFS）

fail指针

与KMP算法中的next数组类似

AC自动机的习题

http://www.cnblogs.com/kuangbin/p/3164106.html

例2 阿狸的打字机（bzoj2434）

阿狸有一台打字机。打字机上有28个按键，分别印有26个小写英文字母和'B'、'P'两个字母。这个打字机是这样工作的：

1. 输入小写字母，打字机的一个凹槽中会加入这个字母(这个字母加在凹槽的最后)。
2. 按一下印有'B'的按键，打字机凹槽中最后一个字母会消失。
3. 按一下印有'P'的按键，打字机会在纸上打印出凹槽中现有的所有字母并换行，但凹槽中的字母不会消失。

例如，阿狸输入aPaPBbP，纸上被打印的字符如下：

a

aa

ab

我们把纸上打印出来的字符串从1开始顺序编号，一直到n。打字机有一个非常有趣的功能，在打字机中暗藏一个带数字的小键盘，在小键盘上输入两个数(x,y)（其中1≤x,y≤n），打字机会显示第x个打印的字符串在第y个打印的字符串中出现了多少次。

例2

首先建出AC自动机。

考虑一个模式串x在另一个模式串y中出现的次数，即是求：整个y在自动机上的这条链的所有节点中，有多少个可以通过fail指针走到x。

按fail指针建出fail树。

问题转化为：求x的子树内，有多少个y链上的节点。

考虑离线，枚举所有模式串作为y，并求所有相关询问的答案。

枚举方法：注意到输入规模不可能太大，可以直接模拟输入，并在AC自动机上走。

求解方法：将当前栈内（屏幕上）的节点维护在一个树状数组内，即可支持快速求解。

Manacher

求以每个位置为中心的回文半径长度

（求偶数长度回文子串的方法：在初始串中相邻两个字符中间插入#）

Manacher的实现

```
int mx = 0,id;
for(int i=1; i<n; i++){
	if( mx > i ) p[i] = MIN( p[2*id-i], mx-i );
	else p[i] = 1;
	for(; str[i+p[i]] == str[i-p[i]]; p[i]++);
	if( p[i] + i > mx ){
		mx = p[i] + i;
		id = i;
	}
}
```

例3 双倍回文

定义双倍回文串：

Wr(W)Wr(W)

其中r(W)为W的倒置。

如abbaabba是双倍回文串，但abaaba不是。

本题中的子串为连续子串。

给定一个串，求最长双倍回文子串的长度。

N<=500000

例3 双倍回文

分析：

1. 首先必须为以#为中心的回文串。
2. 左边/右边的串必须也为回文串。

先跑manacher，求出以每个#为中心的回文半径

然后我们需要求出对于每个i，中心在[i,i+p[i]/2]范围内的，包含i的，中心最右的回文子串（且中心必须为#）。

将所有极长回文子串按左端点排序，从左到右枚举#，利用单调性在set内维护所有左端点小于i的极长回文子串的右端点，每次在set内upper_bound求出符合要求的中心，并更新答案。

后缀数组

sa[i]表示第i名的后缀

rank[i]表示第i个后缀的名次

height[i]表示第i名的后缀和第i-1名的后缀的最长公共前缀

求法：倍增

例4 NOI2015品酒大会

给定一个长度为n的串，每个位置对应了一杯酒。

Str(l,r)表示[l,r]这段的子串。

如果Str(x,x+len-1)=Str(y,y+len-1)，那么称第x杯酒和第y杯酒是len相似的。

每一杯酒有一个美味度a[i]，将两杯酒i,j调到一起，新酒的美味度为a[i]*a[j]。

对于r=0…n-1，分别求找出两杯r相似的酒的方案数，以及选两杯r相似的酒可以得到的最大的美味值。

例4 NOI2015品酒大会

后缀数组裸题

按height数组合并

实现细节：把每个“足够相似”的后缀的集合看成一个对象，重载一些运算符。（需要维护的这个对象的信息：最大值、次大值、最小值、次小值、计数）

后缀自动机

自动机上的每一个节点：

Right集合：一个“右端点”的集合，这个集合中的右端点在一个长度范围意义下的串都相等。

区间：当前节点的Right集合所对应的长度范围

Parent：当前节点的Right集合是Parent的子集。

具体的建法（extend函数）以及其他常见用法需要掌握（可以自行掌握）。

例4 NOI2015品酒大会

后缀自动机裸题

根据每个（自动机）点的区间和right集合大小直接更新答案。

杂题选讲

APIO2014回文串

定义s的一个子串t的出现值为t在s中的出现次数乘以t的长度。

求s的所有回文子串中的最大出现值。

N<=3*10^5

APIO2014回文串

算法1：SAM+manacher

算法2：回文自动机

算法3：hash+manacher

本质不同的回文串之间有父子关系。

本质不同的回文子串是O(n)的。

借助hash建出这棵树，并统计每个点出现的次数。

遍历一遍树上的节点，然后求出答案。

例1 bzoj3790

你有两个机器：一个可以生成一个回文串；另一个可以把两个串拼接起来，特别地，如果（令待拼接的两个串为A,B）A的一个后缀和B的一个前缀相同，那么可以将这个重复部分重叠。

给定一个目标串，求你为了得到这个目标串，至少需要使用多少次第二个机器。

N<=10^5

例1

Manacher求出所有极长回文子串

O(n)扫一遍，贪心即可。

例2 bzoj2081

给定一个串。你可以把这个串分成若干段，每块都有k个字符。（如果最后一段字符数少于k，则丢掉不要）

对于不同的k，你能得到不同的段。两个段是本质相同的，当且仅当它们相等或将其中一个段翻转后它们相等。

你现在想知道：能够获得的最多的本质不同段数；能获得这个最大值的k的数目；能获得这个最大值的所有k。

例2

考虑暴力。

直接枚举k然后求本质不同的hash值。

似乎暴力直接能过？

调和级数求和的量级保证了复杂度！

用set维护出现过的串的hash值。

可能要用双哈希（？）

例3 NOI2016优秀的拆分

如果一个字符串可以被拆分为 AABB 的形式，其中 A 和 B 是任意非空字符串，则我们称该字符串的这种拆分是优秀的。

例如，对于字符串 aabaabaa，如果令 A=aab，B=a，我们就找到了这个字符串拆分成 AABB 的一种方式。

一个字符串可能没有优秀的拆分，也可能存在不止一种优秀的拆分。比如我们令 A=a，B=baa，也可以用 AABB 表示出上述字符串；但是，字符串 abaabaa 就没有优秀的拆分。

现在给出一个长度为 n 的字符串 S，我们需要求出，在它所有子串的所有拆分方式中，优秀拆分的总个数。这里的子串是指字符串中连续的一段。

以下事项需要注意：

出现在不同位置的相同子串，我们认为是不同的子串，它们的优秀拆分均会被记入答案。

在一个拆分中，允许出现 A=B。例如 cccc 存在拆分 A=B=c。

字符串本身也是它的一个子串。

95%的数据：n<=2000；100%的数据：n<=30000

例3

大大暴力（似乎是85分）：

枚举A的位置、长度，B的位置、长度，然后用hash判。

大暴力（似乎是90分）：

枚举A的位置、长度，即可得到B的开始位置，然后再枚举B的长度，然后用hash判断。

暴力（95分）：

优秀的拆分是两个AA形式的拆分拼起来的（两个AA形式的拆分中A可能不同）。不妨先处理出所有AA形式的拆分。

考虑当你确定了B的开始位置，你所关注的仅仅是这个开始位置开始的AA形式的拆分有多少个。

那么我们可以预处理出对于所有i，从第i个位置开始的AA形式的拆分数量f[i]，以及在第i-1个位置结束的AA形式的拆分的数量g[i-1]。这样我们就可以直接计算出答案了。

例3

后缀数组暴力（100分）：

我们注意到，刚刚的算法的瓶颈在于，我们需要枚举出所有AA形式的拆分。我们使用了枚举+hash的方法。现在，我们考虑用后缀数组来做统计。

枚举A的长度L，每L步取一个关键点，那么每个该长度的AA形式的拆分肯定恰好经过两个相邻的关键点。并且，它们位置差距为L的字符一一匹配。

所以可以枚举所有相邻关键点，通过后缀数组求出最长公共前缀、最长公共后缀，即可知道存在多少该长度的AA形式的拆分。

这相当于f和g的一段区间+1，差分前缀和后单点修改即可。

时空间复杂度O(nlogn)。

例4 bzoj2061

有n个字符串变量(n<=26)，它们可以包含其他的字符串变量，也可以包含小写字母。（这些变量用大写字母表示）。

举个栗子：A=greatglorycorrect B=xx C=leadusgo D=ABC E=DDDDdjh F=EEEEEgoodbye

显然，这样的定义必须是无环的。

给定一个模式串，求某一个变量所代表的字符串里这个模式串出现了几次。

模式串长度,每条描述的长度<=100

例4

这个最终的字符串可能会很长很长，所以我们肯定不能求出这个串。

进一步，我们发现，对于一个变量所代表的字符串，我们只关心它的：长度为m的前缀；长度为m的后缀；在这个串内模式串匹配的位置数。（其中m为模式串的长度）

维护这些信息，拓扑排序+合并维护即可。

例5 bzoj3238

一个长度为 n 的字符串 S，令 Ti 表示它从第 i 个字符开始的后缀。求

$$\sum_{1\le i<j\le n} \mathrm{len(Ti)} + \mathrm{len(Tj)} - 2 * \mathrm{lcp(Ti, Tj)}$$

其中，len(a)表示字符串 a 的长度，lcp(a, b)表示字符串 a 和字符串 b 的最长公共前缀。

2<=N<=500000,S由小写英文字母组成

例5

算法1：后缀数组

建出后缀数组。

然后根据height数组从小到大合并即可。

时间O(nlogn)

显然不是最快的。

例5

算法2：后缀自动机

这是一个SAM裸题，根据right集合的定义，以及mx值和节点父亲mx值，可以快速地求解。

会后缀自动机的同学不妨思考一下。

例6 bzoj4453

给定一个串S，有Q个询问，每次问一个区间内字典序最大的子串。

|S|,Q<=10^5

提示：可以离线！

例6 题解

zimpha Oct '15

我是题解的搬运工...先附上新鲜AC的代码, 我比较懒, 直接写了hash, 没用后缀xxxz之类的高级货.

按照右端点从小到大离线处理所有询问. 假设我们现在处理到了端点r, 令ans_i是询问$[i,r]$的答案, 容易发现ans_i显然是一段段相等的数, 并且是单调递增的, 于是我们可以开一个set维护ans_i的值, 对于某个$[l,r]$的答案直接在set里面lower_bound下l就好了.

考虑当$r \to r+1$时, 这个set的变化. 借用叉姐的说法, 对于两个后缀$s[i..n] < s[j..n](i < j)$, 令他们的lcp为l, 那么在r到$j+l$为止, $s[i..r] > s[j..r]$, 我们不妨称i伴随j, 同时当$r = j+l$的时候, 显然i要从这个set里面删掉, 由于一些后缀伴随着i, 那么当i删掉的时候, 那些伴随i的, 以及伴随伴随i的, 都要递归地删掉. 于是当$r \to r+1$时, 具体变化是这样的, 把$i < r+1, s[i..n] < s[r+1..n]$的那些$i$标记一个删除时刻, 同时记$i$伴随$r+1$. 事实上只需要开个单调递增的栈来维护这些后缀, 那么$s[r+1..n]$只需要和栈尾那些小于它的后缀比较就好了, 比较完直接出栈, 因为现在得到的时刻是i最早被删除的时刻, 其他都不是特别重要. 搞完这些后, 把那些删除时刻是$r+1$的删掉, 然后处理右端点等于$r+1$的询问就好了. 复杂度大抵就是$O((n+Q)\log n)$.

Flydutchman, zyeric, TankEngineer, and 3 others like this.

Reply Like

7 bzoj2085

给定n个长度总和不超过1e5的字符串，求一个最短的母串，使所有字符串的出现次数之和=m 这n个字符串保证不互相包含

$n \leq 100$

8 bzoj2803

给定一个字符串S，求一个最长的L(L*2<=n)，使S长度为L的前缀和长度为L的后缀循环同构

9 bzoj3172

给定n个单词，对于每个单词，求其在所有单词中的出现次数总和

10 bzoj4462

给定一个母串和一个模式串，求匹配位置数。

其中小写字母可以作置换。

如AaBaCb可以和AiBiCz匹配，但ABC不能和IJK匹配，aaa也不能和aab匹配

11 bzoj1396

一般地，对于一个字符串 S，和 S 中第 i 个字符 x，定义子串 T=S(i..j)为一个关于 x 的识别子串，当且仅当:

1. i<=x<=j
2. T 在 S 中只出现一次

比如，对于 banana 的第 5 个字符，“nana”，“anan”，“anana”，“nan”，“banan”和“banana”都是关于它的识别子串。

请你写一个程序，计算出对于一个字符串 S，关于 S 的每一位的最短识别子串的长度。

12 bzoj3998

对于一个给定长度为N的字符串，求它的第K小子串是什么。

不同位置的相同子串根据输入决定是否算作一个

FIN