字符串

h10

2018 年 3 月 28 日

目录

目录

NOIP2014 解方程

给定多项式方程

$$\sum_{i=0}^{n} a_i x^i = 0$$

求这个方程在 $[1, m]$ 内的整数解

$0 < n \leq 100, |a_i| \leq 10^6, a_n \neq 0, m \leq 10^6$

对于 70% 的数据有 $m \leq 10000$

NOIP2014 解方程

70% 的做法就是对整个方程放到模意义下多Hash得解

事实发现在 $\bmod p$ 意义下 x 与 $x+p$ 得出的值是相同的，于是我们不需要将 m 个数全都枚举一遍，只需要枚举前 p 个即可

选几个 10000 级别的模数

复杂度 $O(n * \sum p)$

HNOI2014 抄卡组

给你 n 个带通配符 $*$ 的字符串

$*$ 可以通配任意长度个字符（包含0个），问所有字符串是否两两匹配

$n \leq 100000$，输入文件不超过 $10MB$

HNOI2014 抄卡组

如果两个字符串都不包含通配符，直接判断

如果两个字符串都包含通配符，只要Lcp与Lcs的长度都等于到通配符较短的串的长度即可

如果一个包含一个不包含，先执行都包含通配符的操作，再对于带通配符串依次判断其每一部分是否在完整串中顺序出现

NOI2014 动物园

给你一个串 S，定义 $num[i]$ 表示以 i 为结尾的前缀串中，前后缀相等且不相交的长度有多少种，求 S 的 num数组

如 $S = aaaa$ 时 $num[4] = 2$，a 与 aa

$|S| \leq 1000000$

NOI2014 动物园

定义 $next0[i]$：以 i 为结尾的前缀串中，最长的前后缀相等且不相交的长度

先正常求一遍 $next$（从 $next[i-1]$ 沿 $next$ 向上跳）

再求 $next0$（从 $next0[i-1]$ 沿 $next$ 向上跳，直到小于等于 $i/2$）

HNOI2008 GT考试

[题目与题解]

ExKmp

对于两个串文本串 T 与模式串 P，求 T 的每一个后缀与 P 的最长公共前缀，记为 $ex[i]$

要求 $O(n)$

ExKmp

定义 P 的每一个后缀和 P 的最长公共前缀长度为 $next[i]$

假设我们已经得到了 $next$，且 $ex[1]$ 到 $ex[k-1]$ 已得出，现在要求 $ex[k]$

对于前 $k-1$ 次匹配，假设每次匹配的字符串在 T 中是第 s 位到第 t 位

提取出其中 t 最大的一次匹配，并令 $st=s$，令 $ed=t$

因为$T[st,ed]=P[1,ed-st+1]$，所以 $T[st,k-1]=P[1,k-st]$

ExKmp

令 $len = next[k - st + 1]$

情况1: $k + len - 1 < ed$，则 $ex[k] = len$

情况2: $k + len - 1 \geq ed$

这种情况下 $ex[k]$ 至少为 len，然后暴力向后匹配并更新 st 与 ed，由于 ed 不断递增，故暴力匹配次数不超过 n

最后考虑求 $next$ 数组，其求法与 ex 数组十分类似

GDOI2014 beyond

有两个长度为 n 的字符串 A, B，找出最大的 i，使得 A 的前 i 位与 B 的前 i 位可以循环同构

如 $abcdx$，$cdabx$ 两个串答案为 4

$n \leq 2000000$

GDOI2014 beyond

两个串先互相做一遍ExKmp

枚举 A 的每一个后缀，它与 B 的前缀最多能匹配 $exa[i]$ 位

先考虑暴力，从 0 到 $exa[i]$ 枚举 A 的后缀与 B 的前缀匹配掉的长度 len

可以发现答案合法当且仅当 $exb[len+1] >= i-1$，

也就是说我们需要找到 1 到 $exa[i]+1$ 之间最大的 p 满足 $exb[p] >= i-1$，二分+RMQ

Trie例题

给定 n 个数，求这 n 个数两两异或的值中的前 k 小，注意 $a[i] \wedge a[j]$ 与 $a[j] \wedge a[i]$ 是相同的

$n, k \leq 100000, 0 \leq a[i] < 2^{31}$

Trie例题

对于一个数 A，求出 $A \wedge a[i]$ 的第 x 小十分容易，记录Trie每棵子树包含了多少个数即可

一开始把每个数的最小异或值放入一个小根堆中，然后每次取出最小值，把最小值的两个来源的下一小的异或值放入堆中

CF587F

给定 n 个字符串 S_i，每次询问给出 l, r, k，求 $\sum_{i=l}^{r} occur(S_i, S_k)$

其中 $occur(P, T)$ 表示 P 在 T 中的出现次数

$n, q, \sum |S_i| \leq 10^5$

CF587F

因为 n 与 $\sum|S_i|$ 同阶，后文用 n 代替 $\sum|S_i|$

$occur(P,T)$ 相当于fail树中 P 的结束节点的子树内有多少个 T 的结点

情况1: $|S_k|>\sqrt{n}$

对于每一个 S_k 都遍历一遍Trie，这种串不超过 $\sqrt{n}$ 个

情况2: $|S_k|<\sqrt{n}$

首先，

$\sum_{i=l}^{r} occur(S_i,S_k)=\sum_{i=1}^{r} occur(S_i,S_k)-\sum_{i=1}^{l-1} occur(S_i,S_k)$

所以我们只需要求 $\sum_{i=1}^{t} occur(S_i,S_k)$ 即可

CF587F

将询问按 t 从小到大排序，题目变成了动态加入 S_i 并询问 S_k

注意 $occur(P, T)$ 相当于fail树中 P 的结束节点的子树内有多少个 T 的结点

于是对于加入 S_i，把 S_i 的结束点的标记一下

对于询问 S_k，枚举 S_k 在AC自动机上经过的点，并询问它所有祖先的标记次数之和

目前是 $O(n\sqrt{n}\log n)$ 的

CF587F

反过来思考

对于加入 S_i，将 S_i 结束结点在fail树上的子树内所有点的权值加一

对于询问，则可以直接得到答案

把fail树放置在dfs序上

由于修改 $O(n)$ 次，询问 $O(n\sqrt{n})$ 次，故我们可以分块，这样总复杂度还是 $O(n\sqrt{n})$

Manacher

算法实现

Palindromic Tree

算法实现

HEOI2016 str

给定一个字符串 S，有 m 个询问，每组询问形如 (a, b, c, d)，询问 $S[a, b]$ 的所有子串中与 $S[c, d]$ 的最长公共前缀的最大值是多少

$|S|, m \leq 10^5$

HEOI2016 str

首先我们可以二分答案，对于当前答案 len，那么匹配子串在 $S[a,b]$ 中的起点下标为 $[a, b-len+1]$

在SA上确定一段和 $S[c,d]$ 的 LCP 大于等于 len 的区间，利用主席树查询区间中是否有 $[a, b-len+1]$ 中的元素即可

CC TANDEM

给定一个串 S

定义好的子串是形式为 XXX 的子串，即三个相同的串拼起来

定义非常好的子串是满足子串的后面一位与子串开头不同的好的子串，如果子串是一个后缀也算满足

求出 S 中非常好的子串的个数

$|S| \leq 10^5$

CC TANDEM

考虑一个形如 XXX 的子串，我们如果每隔 $|X|$ 设置一个端点，那么这种串必定会经过三个端点

因此考虑枚举 $|X|$ 的大小，考虑如何计算经过连续三个点的满足条件的串的个数

设 $|X| = l$，连续三个端点为 $x, x+l, x+2l$

计算 $LCP(x, x+l, x+2l), LCS(x, x+l, x+2l)$ 就可以算出串开头的范围

对于非常好子串的要求，我们可以发现如果串能往右移一位，那么一定不满足条件，因此只要考虑移到最右边的情况即可

诸神眷顾的幻想乡

给定一棵 n 个结点的树，每个节点上有一个字符

定义路径为某两点间的节点上的字符构成的字符串（有向）

求树上有多少条互不相同的路径

$n \leq 10^5$，叶子结点个数不超过 20 个

诸神眷顾的幻想乡

考虑叶子结点不多，那么以每个叶子为根建一棵Trie树并且合并，那么原树每条路径都可以表示成从根出发自上而下的路径了

然后建出这棵Trie树的SAM，统计不同字符串个数即可

CC TASUFFIX

设数组 $A[i] = i$，大小为 n，对其进行 m 次操作

有两种操作：

1. 把某一段提到开头
2. 区间翻转

所有操作结束后询问后缀数组为 A 的字符串 S 有多少种可能

S 的限制是字符串中出现的都是正整数，且最大元素等于不同的元素个数

$n \le 10^9, m \le 10^5$

CC TASUFFIX

我们来考虑一下题意：$suf[A_i, n] < suf[A_{i+1}, n]$

考虑到 S 的限制，则一定有 $S[A_i] \leq S[A_{i+1}]$

当上式取小于号时，一定可行，现在主要是要看能不能取等于号

发现等于号取到的条件就是 $suf[A_i + 1, n] < suf[A_{i+1} + 1, n]$

CC TASUFFIX

用Splay来得到最终的 A 数组，其形式必定是：

$[x_1, y_1], [x_2, y_2], \ldots, [x_k, y_k]$

也就是一段段连续的区间

我们发现当 A_i, A_{i+1} 不在区间边界上的时候是一定可以满足取等的条件的，因此只需要考虑边界上是否满足条件即可，这个也很容易

最后的答案就是 2^p，其中 p 为可以取等的位置数

Thank

Thank you for listening