字符串相关算法

Wearry

Jan 13, 2020

Table of Contents

NOIP 2014 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 (保证存在)。

$n \leq 100, |a_i|, m \leq 10^6$

NOIP 2014 解方程

注意到如果这个方程左边取到 0，那么在 $(\bmod p)$ 的意义下一定也等于 0，因此可以选取若干个不同的质数 $\{p_i\}$，使用枚举的方式求出模意义下的解，最后合并即可得到答案。

注意到在考虑 $(\bmod p)$ 的情形时，不需要枚举 $[1, m]$ 中的所有数，只需要考虑 $[1, p]$ 即可。

这样的复杂度是 $O(n \sum_i p_i)$ 的。

HNOI2014 抄卡组

给出 n 个带有通配符 $*$ 的字符串，其中 $*$ 可以匹配任意个字符 (包括 0)，现在你需要回答任意两个字符串之间能否相互匹配。

$n \leq 10^5$，输入大小不超过 10MB

HNOI2014 抄卡组

由于通配符具有非常强的性质，因此我们不妨分情况考虑：

- 如果两个字符串都不包含通配符，直接判断即可。

HNOI2014 抄卡组

由于通配符具有非常强的性质，因此我们不妨分情况考虑：

- 如果两个字符串都不包含通配符，直接判断即可。
- 如果两个字符串都包含通配符，只需要 $\mathrm{LCP}, \mathrm{LCS}$ 的长度都等于距离通配符较短的串的长度即可。

HNOI2014 抄卡组

由于通配符具有非常强的性质，因此我们不妨分情况考虑：

- 如果两个字符串都不包含通配符，直接判断即可。
- 如果两个字符串都包含通配符，只需要 $\mathrm{LCP}, \mathrm{LCS}$ 的长度都等于距离通配符较短的串的长度即可。
- 如果其中一个包含通配符，另一个不包含，则需要判断包含通配符的串的每一个部分是否完整的按照顺序在另一个串中出现过。

NOI2014 动物园

给你一个串 S，定义 num_i 表示以 i 结尾的前缀串中，前后缀相等且不相交的长度有多少种，求所有的 num_i。

$|S| \leq 1000000$

NOI2014 动物园

定义 $next0_i$ 表示以 i 为结尾的前缀串中，最长的前后缀相等且不相交的长度。

首先正常的做一遍 KMP 求出 $next$，然后利用 $next$ 求出的结果求出 $next0$。

NOI2014 动物园

定义 $next0_i$ 表示以 i 为结尾的前缀串中，最长的前后缀相等且不相交的长度。

首先正常的做一遍 KMP 求出 $next$，然后利用 $next$ 求出的结果求出 $next0$。

注意到在求 $next_i$ 的时候是先从 $next_{i-1}$ 开始的，求 $next0_i$ 时类似的从 $next0_{i-1}$ 开始即可。

HNOI2008 GT 考试

求所有长度为 n 的数字串中不包含给定的长度为 m 的模式串的数量，对 k 取模。

$n \leq 10^9, m \leq 20$

HNOI2008 GT 考试

考虑使用 DP 的方式计算答案，记 $f_{i,j}$ 表示数字串的前 i 位已经确定，并且最长的后缀能够匹配模式串的一个前缀的长度为 j 的方案数，转移的具体过程可以由 KMP 确定。

HNOI2008 GT 考试

考虑使用 DP 的方式计算答案，记 $f_{i,j}$ 表示数字串的前 i 位已经确定，并且最长的后缀能够匹配模式串的一个前缀的长度为 j 的方案数，转移的具体过程可以由 KMP 确定。

又注意到对于不同的 i 来说，转移的系数都是固定的，因此可以利用矩阵乘法优化。

ExKMP

对于文本串 T 与模式串 P，求 T 的每一个后缀与 P 的 LCP，记为 ex_i。

要求做到 $O(n)$。

ExKMP

- 定义 P 的每一个后缀与 P 本身的 LCP 为 len_i，接下来考虑如何从小到大求出所有的 len 来。

ExKMP

- 定义 P 的每一个后缀与 P 本身的 LCP 为 len_i，接下来考虑如何从小到大求出所有的 len 来。
- 假设我们已经知道了 len_1 到 len_{i-1}，现在要求 len_i，不妨记之前 $j+len_j$ 的最大值在 $t, t+len_t$ 取到。

ExKMP

- 定义 P 的每一个后缀与 P 本身的 LCP 为 len_i，接下来考虑如何从小到大求出所有的 len 来。
- 假设我们已经知道了 len_1 到 len_{i-1}，现在要求 len_i，不妨记之前 $j+len_j$ 的最大值在 $t, t+len_t$ 取到。
- 于是我们可以用 len_{i-t} 来更新 len_i 的初始值，对于剩下的部分暴力求出即可。

ExKMP

- 定义 P 的每一个后缀与 P 本身的 LCP 为 len_i，接下来考虑如何从小到大求出所有的 len 来。
- 假设我们已经知道了 len_1 到 len_{i-1}，现在要求 len_i，不妨记之前 $j+len_j$ 的最大值在 $t, t+len_t$ 取到。
- 于是我们可以用 len_{i-t} 来更新 len_i 的初始值，对于剩下的部分暴力求出即可。
- 不难发现，这个过程中 $j+len_j$ 的最大值一定单调，因此复杂度是线性的。

ExKMP

- 定义 P 的每一个后缀与 P 本身的 LCP 为 len_i，接下来考虑如何从小到大求出所有的 len 来。
- 假设我们已经知道了 len_1 到 len_{i-1}，现在要求 len_i，不妨记之前 $j+len_j$ 的最大值在 $t, t+len_t$ 取到。
- 于是我们可以用 len_{i-t} 来更新 len_i 的初始值，对于剩下的部分暴力求出即可。
- 不难发现，这个过程中 $j+len_j$ 的最大值一定单调，因此复杂度是线性的。
- 对于原问题，可以利用类似的方式求解。

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 位循环同构。

如 abcdx，cdabx 答案为 4。

$n \leq 2 \times 10^6$

GDOI2014 Beyond

首先两个串互相做一次 ExKMP，得到两个数组 $lena_i, lenb_i$，分别表示 A, B 的每一个后缀与 B, A 之间最长公共前缀的长度。

GDOI2014 Beyond

首先两个串互相做一次 ExKMP，得到两个数组 $lena_i, lenb_i$，分别表示 A, B 的每一个后缀与 B, A 之间最长公共前缀的长度。

接下来首先枚举 A 的一个后缀 $A[i,n]$，然后不妨假设它与 B 的匹配长度为 L。

那么这个方案合法当且仅当 $lenb_{L+1} \geq i-1$，二分然后 RMQ 即可。

例题：Trie 的应用

给定 n 个数，求这 n 个数两两异或的值中的前 k 小。

$n, k \leq 10^5, 0 \leq a_i \leq 2^{31}$

例题：Trie 的应用

对于一个数 X，求出 $X \oplus a_i$ 的第 k 小非常简单，记录 Trie 的每棵子树中插入的数的总个数即可。

例题：Trie 的应用

对于一个数 X，求出 $X \oplus a_i$ 的第 k 小非常简单，记录 Trie 的每棵子树中插入的数的总个数即可。

对于两两异或的情况，可以利用优先队列来进行优化，首先将每个 a_i 对应的最小异或值插入优先队列中，每次取出队列中的最小值之后，利用 Trie 得到下一个可能成为最小值的数即可。

CF587 F

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 $occur(S, T)$ 表示 S 在 T 中出现的次数。

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

CF587 F

首先将问题进行转化，$occur(S,T)$ 实际上是 AC 自动机上 T 的所有前缀对应的点在 fail 树中 S 对应的点的子树中出现的次数。

CF587 F

首先将问题进行转化，$occur(S,T)$ 实际上是 AC 自动机上 T 的所有前缀对应的点在 fail 树中 S 对应的点的子树中出现的次数。

分情况讨论：

1. $|S_k| > \sqrt{n}$，这样的 k 不超过 $\sqrt{n}$ 个：对于每个 S_k，在树上遍历一遍预处理前缀和。

CF587 F

首先将问题进行转化，$occur(S,T)$ 实际上是 AC 自动机上 T 的所有前缀对应的点在 fail 树中 S 对应的点的子树中出现的次数。

分情况讨论：

1. $|S_k| > \sqrt{n}$，这样的 k 不超过 $\sqrt{n}$ 个：对于每个 S_k，在树上遍历一遍预处理前缀和。
2. $|S_k| \leq \sqrt{n}$：同样只需要计算前缀和即可，每次加入 S_i 时将 S_i 对应的子树权值加一，询问时枚举 S_k 的所有前缀节点得到答案。

复杂度可以做到 $O(n\sqrt{n})$。

Manacher & Palindromic Tree

Manacher

需要解决的问题是求出每个位置的最长回文半径，记一个 len_i 来表示这个东西，实现的方式和复杂度的分析都与 ExKMP 类似。

Manacher & Palindromic Tree

Manacher

需要解决的问题是求出每个位置的最长回文半径，记一个 len_i 来表示这个东西，实现的方式和复杂度的分析都与 ExKMP 类似。

Palindromic Tree

求出一个树形结构，包含串中所有的回文子串，同时包含 fail 属性，表示最长回文前后缀。

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 LCP 最大值。

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

HEOI2016 Str

首先可以二分答案：

- 假设二分的答案是 L，那么匹配的子串在 $S[a,b]$ 中起点的范围就是 $[a,b-L+1]$。

HEOI2016 Str

首先可以二分答案：

- 假设二分的答案是 L，那么匹配的子串在 $S[a,b]$ 中起点的范围就是 $[a,b-L+1]$。
- 在后缀数组上满足和串 $S[c,d]$ 的 LCP 大于等于 L 的部分形成一段区间，利用主席树查询区间中是否包含 $[a,b-L+1]$ 的后缀即可。

CC TANDEM

定义一个串是好的当且仅当它能表示成 XXX 的形式，即三个相同的字串按顺序连接起来。定义一个串是非常好的当且仅当它是好的同时它的结尾的后一个字符与起始字符不同。

给定一个串 S，求它的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，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，然后计算经过连续三个点的满足条件的串的个数。

CC TANDEM

考虑一个形如 XXX 的子串，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，然后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 $LCP(x, x+l, x+2l)$ 和 $LCS(x, x+l, x+2l)$ 就可以得到串开头的范围。

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 个节点的树，每个节点上都恰好有一个字符，定义路径为某两个点之间的所有字符按顺序构成的字符串，求树上有多少条互不相同的路径。

$n \leq 10^5$，叶子节点的数量不超过 20。

诸神眷顾的幻想乡

由于叶子结点的数量不多，考虑以每个叶子为根分别建一棵 Trie，那么树上的每一条路径都能表示为 Trie 上某一个串的子串。

诸神眷顾的幻想乡

由于叶子结点的数量不多，考虑以每个叶子为根分别建一棵 Trie，那么树上的每一条路径都能表示为 Trie 上某一个串的子串。

考虑建立广义的后缀自动机，在广义后缀自动机上将每棵 Trie 包含的字符串插入，直接利用后缀自动机本身的性质统计出不同字符串的个数即可。

CC TASUFFIX

设有一长度为 n 的序列，初始时满足 $a_i = i$，接下来有 m 次操作：

1. 将某一段提到开头
2. 区间翻转

在所有操作结束后询问后缀数组为 $\{a_i\}$ 的字符串 S 的方案数，要求 S 中出现的字符均为正整数，且最大元素等于元素种类数。

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

CC TASUFFIX

考虑后缀数组的意义:

$$suf[a_i, n] < suf[a_{i+1}, n]$$

一定有 $S[a_i] \leq S[a_{i+1}]$，如果取小于号，则一定满足条件，否则问题转化成:

$$suf[a_i + 1, n] < suf[a_{i+1} + 1, n]$$

CC TASUFFIX

用平衡树维护所有操作得到序列 $\{a_i\}$，一定会得到若干连续的区间，发现 a_i, a_{i+1} 不在区间边界上的时候一定能够取等号，否则只需特殊判断即可。

假设最后求出的能够取等号的位置有 p 个，答案就是 2^p。