String Algorithms

wlxhkk

2023 年 7 月 12 日

引言

pdf 是抄 Charlie 的。

CF196D The Next Good String

Statement

- 给定长度为 n 的小写字母串 s 和正整数 d。
- 定义一个串是好的当且仅当它没有长度大于等于 d 的回文子串。
- 请你求出长度为 n 并且字典序比 s 严格大的好串里，字典序最小的那个，或报告无解。
- $1 \le d \le n \le 4 \times 10^5$。

CF196D The Next Good String

Solution

- 很简单的一道题。首先要找出第一个变化的位置，注意到有长为 L 的回文子串意味着有长为 $L-2$ 的回文子串，因此只需要对 d 和 $d+1$ 判定即可。

- 很简单的一道题。首先要找出第一个变化的位置，注意到有长为 L 的回文子串意味着有长为 $L-2$ 的回文子串，因此只需要对 d 和 $d+1$ 判定即可。
- 因此找到第一个不合法回文子串的末尾，从这个位置开始往前逐个增加每个位置上的字符，直到末尾不是不合法的回文子串。从此位置再向后贪心填数即可，显然此时至多只有 2 个不合法值。
- 判断回文可以哈希。

- 称一个字符串是好的，当且仅当它是一个长度为偶数的回文串或由若干长度为偶数的回文串拼接而成。给定一个长度为 n 的小写字母组成的字符串 s，求有多少 s 的子串是好的。
- $1 \leq n \leq 5 \times 10^5$。$1\text{s}, 256\text{MB}$。

- 建一个 $n+1$ 个点的图。对于一个偶回文串 (l,r) 连边 $l-1 \to r$，那么 $s[l:r]$ 合法等价于存在 $l-1$ 到 r 的路径。

- 建一个 $n+1$ 个点的图。对于一个偶回文串 (l,r) 连边 $l-1 \to r$，那么 $s[l:r]$ 合法等价于存在 $l-1$ 到 r 的路径。
- 进一步的，发现如果存在边 $(v,u),(w,u),v<w$ ，由回文串性质可知同样存在 v 到 w 的路径。因此可以将有向边换成无向边，求出所有联通块计算。先跑一遍 Manacher 算法，然后用 [SCOI2016] 萌萌哒的方法连边即可。

CF1090J Two Prefixes

Statement

- 给定两个字符串 S、T，求 T 的非空前缀拼接在 S 的非空前缀后可以得到的本质不同字符串数。
- $|S|, |T| \leq 10^7$。

CF1090J Two Prefixes

Solution

- 先把答案设为 $|S| \cdot |T|$ ，然后去除重复计算的部分。
- 假设串 $c = a + b = a' + b'$ ，其中 a, a' 为 s 的前缀，b, b' 为 t 的前缀，$|a| < |a'|$ 。我们希望只将最短的 a 计入答案。

- 先把答案设为 $|S| \cdot |T|$ ，然后去除重复计算的部分。
- 假设串 $c = a + b = a' + b'$ ，其中 a, a' 为 s 的前缀，b, b' 为 t 的前缀，$|a| < |a'|$ 。我们希望只将最短的 a 计入答案。
- 那么容易发现，b' 是 b 的 border ，那么考虑有多少个 a' 需要减掉。对于一个前缀 b ，我们钦定 b' 为其最长的 border 并除去其对应的 a'。容易发现这样不会算重。

- 先把答案设为 $|S|\cdot|T|$ ，然后去除重复计算的部分。
- 假设串 $c=a+b=a'+b'$ ，其中 a,a' 为 s 的前缀，b,b' 为 t 的前缀，$|a|<|a'|$ 。我们希望只将最短的 a 计入答案。
- 那么容易发现，b' 是 b 的 border ，那么考虑有多少个 a' 需要减掉。对于一个前缀 b ，我们钦定 b' 为其最长的 border 并除去其对应的 a'。容易发现这样不会算重。
- 那么，我们只需要减去 $b-b'$ 在 S 中出现的次数即可。可以将 S 与 T 跑 KMP ，对每个位置记录其（及其 border ）被匹配了几次。
- 需要对空前缀进行一点处理。时间复杂度 $O(n)$ 。

- 给定一个操作序列 t ($t_i \in [\mathtt{a}, \mathtt{z}] \cup \{\mathtt{P}, \mathtt{B}\}$)，按照操作序列维护一个当前字符串 s。
- 有三种操作：末尾新加一个字符；删除末尾字符；打印当前字符串。
- m 次询问控制台上第 x 行字符串在第 y 行字符串中出现几次。
- $1 \leq |t|, m \leq 10^5$。$1\text{s}, 128\text{MB}$。

- 应该放在 T1 的。

- 应该放在 T1 的。
- 发现三种操作都可以描述成 trie 树上的游走，因此我们可以建出 AC 自动机。
- 询问相当于问一个串的节点在另一个串的结束节点的 fail 树子树中出现了几次。用树状数组维护即可。时间 $O(n\log n)$。

- 给定若干个模式串 $\{s_n\}$，求是否存在一个无限长的字符串使得其中不包含任何模式串。
- $1 \le n \le 2000$，$\sum |s_i| \le 3 \times 10^4$。$1\text{s}, 128\text{MB}$。

P2444 [POI2000] 病毒

Solution

- 建 AC 自动机，并且标记模式串的终止节点，那么存在无限长的合法串当且仅当存在一个不经过标记点的无限长路径。从根开始 DFS ，用 vis 数组判环即可。

- 给出一个包含 n 个词的字典 $S_1, S_2, \ldots, S_n$，求由长度为 m 的字符串 S 和划分 P, Q 组成的三元组 (S, P, Q) 数量，满足：
  - 对于 P 中划分的每一段 $[l, r]$，均满足 $S[l, r]$ 在字典中。
  - 对于 Q 中划分的每一段 $[l, r]$，均满足 $S[l, r]$ 在字典中。
- 答案对 998244353 取模。$n \le 8, m \le 10^9, |S_i| \le 5$。

CF1511F Chainword

Solution

- 发现 $\sum |s| \leq 40$ ，考虑在 trie 树上 DP 。记 $f_{i,u,v}$ 表示长为 i 的字符串，P 匹配到结点 u ，Q 匹配到结点 v 的方案数。发现 i 这一维可以用矩乘优化，但矩阵大小是 40^2 的，无法接受。

- 发现 $\sum |s| \le 40$ ，考虑在 trie 树上 DP 。记 $f_{i,u,v}$ 表示长为 i 的字符串，P 匹配到结点 u ，Q 匹配到结点 v 的方案数。发现 i 这一维可以用矩乘优化，但矩阵大小是 40^2 的，无法接受。
- 注意到 u 是 v 的后缀，或 v 是 u 的后缀。则将无用的状态去除，并只保留 $u \le v$ 的状态，最后只剩下 200 个左右，足以通过。不卡常。

P4548 [CTSC2006] 歌唱王国

Statement

- 有一个无限长的字符串 s，每个字符都是 $[1,n]$ 的均匀随机整数。T 组数据，每次给出字符串 t，求 s 的最短的包含 t 的前缀长度期望。
- $1 \le n, |t| \le 10^5$，$T \le 50$。$3\text{s}, 256\text{MB}$。

- 对该串建出 AC 自动机（KMP 自动机），考虑在 AC 自动机上 DP。设 $E(u)$ 表示 u 节点走到终止节点的期望步数。则有：

$$E(u) = \frac{E(nt_u) + \sum E(v)}{n} + 1$$

即 $E(nt_u) = nE(u) - \sum E(v) - n$，其中 v 是 u 向前指的节点，nt_u 是 u 的下一个节点。显然共有 $n-1$ 个 v。

- 对该串建出 AC 自动机（KMP 自动机），考虑在 AC 自动机上 DP。设 $E(u)$ 表示 u 节点走到终止节点的期望步数。则有：

$$E(u) = \frac{E(nt_u) + \sum E(v)}{n} + 1$$

即 $E(nt_u) = nE(u) - \sum E(v) - n$ ，其中 v 是 u 向前指的节点，nt_u 是 u 的下一个节点。显然共有 $n-1$ 个 v。

- 不妨设 $E(0) = x$ ，可以归纳证明 $E(i) = x - f_i(n)$ ，其中 $f_i(n)$ 是一个和 n 有关的函数。

- 对该串建出 AC 自动机（KMP 自动机），考虑在 AC 自动机上 DP。设 $E(u)$ 表示 u 节点走到终止节点的期望步数。则有：

$$E(u) = \frac{E(nt_u) + \sum E(v)}{n} + 1$$

即 $E(nt_u) = nE(u) - \sum E(v) - n$ ，其中 v 是 u 向前指的节点，nt_u 是 u 的下一个节点。显然共有 $n-1$ 个 v。
- 不妨设 $E(0) = x$ ，可以归纳证明 $E(i) = x - f_i(n)$ ，其中 $f_i(n)$ 是一个和 n 有关的函数。
- 手玩几个发现 $f_i(n) = \sum_{a[1,k]=a[i-k+1,i]} n^k$ ，接下来考虑归纳证明。
- 考虑 $E(i)$ 推到 $E(i+1)$ 的过程，$nE(i) - n$ 的含义是将 $E(i)$ 的 border 后接一个字符后贡献给 $E(i+1)$ 。然而，我们只需要下一个字符 $= t_{i+1}$ 的 border。那么 v_c 指向的点是最长的后接上字符 c 的 border 。由于 border 的 border 也是 border，因此所有不是 border 的串都会被去除。

- 对该串建出 AC 自动机（KMP 自动机），考虑在 AC 自动机上 DP。设 $E(u)$ 表示 u 节点走到终止节点的期望步数。则有：

$$E(u)=\frac{E(nt_u)+\sum E(v)}{n}+1$$

即 $E(nt_u)=nE(u)-\sum E(v)-n$ ，其中 v 是 u 向前指的节点，nt_u 是 u 的下一个节点。显然共有 $n-1$ 个 v。
- 不妨设 $E(0)=x$ ，可以归纳证明 $E(i)=x-f_i(n)$ ，其中 $f_i(n)$ 是一个和 n 有关的函数。
- 手玩几个发现 $f_i(n)=\sum_{a[1,k]=a[i-k+1,i]} n^k$ ，接下来考虑归纳证明。
- 考虑 $E(i)$ 推到 $E(i+1)$ 的过程，$nE(i)-n$ 的含义是将 $E(i)$ 的 border 后接一个字符后贡献给 $E(i+1)$ 。然而，我们只需要下一个字符 $=t_{i+1}$ 的 border。那么 v_c 指向的点是最长的后接上字符 c 的 border 。由于 border 的 border 也是 border，因此所有不是 border 的串都会被去除。
- 因为 $E(n)=0$ ，所以 $ans=x=f_{|t|}(n)$ 。时间复杂度 $O(n)$。

- 给定一个长度为 n 字符串 s 和一个询问参数 k。m 次询问 (l, r)，求 $\sum_{l \le i \le j \le r} [\mathrm{lcp}(s[i:n], s[j:n]) \ge k]$。
- $1 \le n \le 3 \times 10^6$，$1 \le m \le 10^5$，$\boldsymbol{n^2 m \le 10^{15}}$。$3\mathrm{s}, 512\mathrm{MB}$。

P5112 FZOUTSY

Solution

- 显然只关心每个后缀的前 k 个字符，于是求出每个后缀前 k 个字符的哈希值，i, j 的 $\operatorname{lcp} \geq k$ 就相当于哈希值相等。使用莫队做到 $\mathcal{O}(n\sqrt{m})$。

- 给定一个长度为 n 的字符串 s 和 k_1, k_2，求

$$\sum_{1\le i<j\le n} \operatorname{lcp}(i,j)\operatorname{lcs}(i,j)[\operatorname{lcp}(i,j)\le k_1][\operatorname{lcs}(i,j)\le k_2]$$

- $1\le n\le 10^5, 1\le k_1, k_2\le n$。$2\mathrm{s}, 512\mathrm{MB}$。

- 对于每一对 (i,j)，都满足 $(i-lcp(i,j), i+lcs(i,j))$ 和 $(j-lcp(i,j), j+lcs(i,j))$ 这两段区间相等，且是极长的，这启发我们找到所有的极长相等连续段对来计算答案。

- 对于每一对 (i,j)，都满足 $(i-lcp(i,j), i+lcs(i,j))$ 和 $(j-lcp(i,j), j+lcs(i,j))$ 这两段区间相等，且是极长的，这启发我们找到所有的极长相等连续段对来计算答案。
- 对于一对长度均为 len 的极长相等连续段，它的贡献就是 $\sum\limits_{i=1}^{len} i(len-i+1)[i \le k2][len-i+1 \le k1]$，即枚举前缀的长度，容易 $\mathcal{O}(1)$ 计算且预处理出来。

- 对于每一对 (i,j)，都满足 $(i-lcp(i,j), i+lcs(i,j))$ 和 $(j-lcp(i,j), j+lcs(i,j))$ 这两段区间相等，且是极长的，这启发我们找到所有的极长相等连续段对来计算答案。
- 对于一对长度均为 len 的极长相等连续段，它的贡献就是 $\sum_{i=1}^{len} i(len-i+1)[i \le k2][len-i+1 \le k1]$，即枚举前缀的长度，容易 $\mathcal{O}(1)$ 计算且预处理出来。
- 之后的问题在于如何枚举极长相等连续段对，考虑在 SAM 的 parent 树上计算。每次合并两个儿子，左边不能扩展的条件相当于长度取到当前节点的 maxlen 值，右边不能扩展就需要两边的 endpos 下一个位置的字符不同。所以开个桶 $cnt[u][c]$ 表示 u 节点的所有 endpos 集合中，下一个字符为 c 的个数。时间复杂度 $\mathcal{O}(n|\Sigma|)$。

感谢聆听！