字符串专题

A_zjzj 郑钧

北京大学

2026 年 2 月 25 日

一些基础

- Hash 字符串哈希
- $O(nm)$ 求最长公共子序列（Longest Common Subsequence）
- Trie 字典树
- kmp 匹配 / 求 Border

2019 CCPC Final C. Mr. Panda and Typewriter

题目描述

题意：给出长度为 n 的字符串 S，你有两个字符串 P, Q，初始为空，你可以进行如下操作：

- 在 P 的末尾添加任意一个字符，代价是 X；
- 将 Q 赋值为 P 的任意一个子串，代价是 Y；
- 在 P 的末尾拼接上 Q，代价是 Z。

你需要使用最小的代价使得 P 和 S 相同，求最小代价。

$1 \le n \le 5000$，字符集为 $[1, 10^9]$。

解法思路

- dp，设 $f_{i,j}$ 表示当前 $P = S[1:i], Q = S[i-j+1:i]$ 的最小代价，g_i 表示 $P = S[1:i]$ 的最小代价。
- $f_{i,j}$ 的转移：
  - 如果当前的 Q 还有用的话，那么一定是找下一个最近的匹配的子串，对于所有 i, j 可以用单调栈 $O(n^2)$ 预处理出来。
  - 否则没用的话直接从 $f_{i,j}$ 转移到 g_i。
- g_i 的转移，要么下一个插入一个字符；要么下一个插入一个子串，只需要枚举子串长度 j 判断 $S[i+1:i+j]$ 是否是 $S[1:i]$ 的子串，转移到 $f_{i+j,j}$。
- 时空复杂度：$O(n^2)$。

2022 ICPC Hangzhou Regional K. Master of Both

题目描述

给定 n 个仅包含小写字母的字符串 $s_1, \cdots, s_n$，q 次询问，每次询问给出一个 a-z 的字典序，你需要求出，在这个字典序下的 $s_1, \cdots, s_n$ 的逆序对数量。

$1 \le n \le 5 \times 10^5$，$1 \le q \le 5 \times 10^4$，$\sum |s_i| \le 10^6$。

解法思路

- 考虑两个串，要么是前缀关系，此时对答案的贡献是定值；要么不是，此时对答案的贡献仅和出现不同的那一位的两个字符有关。
- 考虑从前往后依次把每个穿插入 Trie 中，并维护前缀关系的贡献，和一个 26×26 的贡献表。
- 查询时遍历贡献表即可。
- 时间复杂度：$O(26 \sum |s_i| + 26^2 q)$。

2022 CERC H. Insertions

题目描述

给定三个字符串 s, t, p，把 t 插入到 s 中共有 $|s|+1$ 种插入位置，你需要对于每个插入位置，计算插入完成后 p 在新的 s 中的出现次数。$1 \le |s|, |t|, |p| \le 10^5$。

解法思路

- 设新的串为 $u_i + t + v_i$，其中 u_i, v_i 为 s 分成前后 $i, |s| - i$ 的两个串。
- p 的出现可以分六种情况讨论（本质上只有三种）：分别独立出现在 u_i, t, v_i 中；跨 u_i, t 或 t, v_i 出现；跨三个串出现。
- 第一种情况直接 kmp 即可解决，第二种情况可以预处理可以产生贡献的前缀/后缀，再做一次 kmp 匹配解决。
- 第三种情况就是一个两维都是树上偏序的二维数点，一维 dfs，另一位树状数组即可。

CF547E Mike and Friends

给定 n 个字符串 $s_1, \cdots, s_n$，q 次询问，每次询问给出 l, r, k，求出 s_k 在 $s_l, s_{l+1}, \cdots, s_r$ 中出现次数之和。
$1 \le n, \sum |s_i| \le 2 \times 10^5$，$1 \le q \le 5 \times 10^5$。

解法思路

- 考虑用 AC 自动机完成匹配。
- 如果两个节点 u, v，在 fail 树上 u 是 v 的祖先，那么 u 表示的串是 v 表示的串的后缀。
- 所以考虑所有 $s_l, \cdots, s_r$ 的前缀表示的串，只需要计算 s_k 表示的串在 fail 树的子树中所有 $s_l, \cdots, s_r$ 的前缀的出现次数即可。

CF587F Duff is Mad

题目描述

给定 n 个字符串 $s_1, \cdots, s_n$，q 次询问，每次询问给出 l, r, k，求出 $s_l, s_{l+1}, \cdots, s_r$ 在 s_k 中出现次数之和。
$1 \le n, \sum |s_i|, q \le 10^5$。

解法思路

- 仍然沿用前一题的思路，用 AC 自动机完成匹配。
- 需要对于 s_k 的每个前缀 u 求出有多少个 $s_l, \cdots, s_r$ 的串在 fail 树上是 u 的祖先。
- 考虑分块，对于 $|s_k| > \sqrt{n}$ 的情况，$O(\sum |s_i|)$ 处理出 $s_1, \cdots, s_n$ 在 s_k 中的出现次数，每次查询求出 $[l, r]$ 的区间和即可。
- 对于 $|s_k| \le \sqrt{n}$ 的情况，对于每个前缀 u，求出有多少个 $s_l, \cdots, s_r$ 的串在 fail 树上是 u 的祖先。这样至多有 $q\sqrt{n}$ 个询问，n 个修改，离线下来在 fail 树上 dfs 做一遍，使用分块做到 $O(\sqrt{n})$ 修改 $O(1)$ 查询即可。
- 总时间复杂度 $O((\sum |s_i| + q + n)\sqrt{n})$。

洛谷 P10716【MX-X1-T4】「KDOI-05」简单的字符串问题

题目描述

给定字符串 S。q 个询问，每次给出 i, k，求有多少个非空字符串 A，使得存在可空字符串 $B_1, B_2, \ldots, B_{k-1}$ 满足：

$$S[1, i] = AB_1AB_2A \ldots AB_{k-1}A$$

对于 100% 的数据：$1 \le k \le i \le n \le 2 \times 10^5$，$1 \le q \le 2 \times 10^5$，$S$ 仅包含小写字母。

解法思路

- 首先特判 $k = 1$ 的情况。
- 剩余的情况，显然 A 是 $S[1:i]$ 的 Border。并且，如果 A 可行，那么 A 的所有 Border 都可行。所以我们只需要求出最长的那个 A 即可。
- 判断 $S[1:i]$ 的一个 Border A 是否可行时，只需要判断 A 在 $S[1:i]$ 最多的不重复的出现次数是否 $\geq k$。
- 考虑预处理 $p_{i,j}$ 表示 $A = S[1:i]$ 时，不重复的第 j 个 A 的末尾在哪里，由于 $j \leq \frac{n}{i}$，因此 p 的是 $O(n \log n)$ 量级的。
- 更进一步地，求出 $p_{i,j}$ 时，我们需要求出 A 在 $j+1$ 开始第一次出现的位置。

解法思路

- 先做一遍 exkmp 求出 z 数组。
- 从小到大枚举 i，那么 $p_{i,j+1} - i + 1$ 的值就是满足 $k \geq p_{i,j} + 1, z_k \geq i$ 的最小的 k。
- 由于 i 从小到大，每次删掉 $z_k < i$ 的位置，那么只需要求 $\geq p_{i,j} + 1$ 第一个没有被删除的位置，使用并查集即可。
- 最后，查询的时候只需要倍增地跳 Border，判断当前 $A = S[1:x]$ 时是否可行即可。
- 总时间复杂度：$O(n \log n)$。

2026 THUPC 初赛 B. 回响形态

题目描述

给定一个长度为 n 的小写字母串，称子串 $s[i:j]$ 的中心为 $\frac{i+j}{2}$。
q 次询问，每次询问给出 k，求出 s 的所有中心为 $\frac{k}{2}$ 的子串的 Border 个数之和。在此题的定义中，一个串本身也是自身的 Border。
$1 \le n \le 10^6$，$1 \le q \le 20$，$2 \le k \le 2n$，0.5s。

解法思路 1 - 官方解法

- 不妨设 $k = n + 1$，考虑串 $t = s_1 s_n s_2 s_{n-1} \cdots s_{n-1} s_2 s_n s_1$。
- 容易发现，原串的中心为 $\frac{n+1}{2}$ 的 Border 恰好与 t 中一个起始下标为奇数的偶回文串一一对应。
- 使用 Manacher 即可做到 $O(qn)$。

解法思路 2 - 后缀数组

- 考虑如果 u 是 $s[l:r]$ 的 Border，那么 u 掐头去尾得到的串一定是 $s[l+1:r-1]$ 的 Border。
- 那么反过来，只需要从每个 u 的长度为 $1,2$ 的情况开始，向左向右求出能够匹配多长，计算贡献即可。
- 匹配的事情只需要正反串各做一边 SA 即可，如果 SA 实现得不好，会被卡常。
- 时间复杂度是 $O(qn+n\log n)$，空间复杂度是 $O(n\log n)$。

SP1811 LCS - Longest Common Substring

题目描述

给定两个字符串 S, T，求出 S, T 的最长公共子串的长度。

$1 \leq |S|, |T| \leq 2.5 \times 10^5$。

解法思路

- 使用字母表之外的一个字符连接 S, T 后做一遍 SA。
- 如果 $\mathtt{sa}_{i-1}$ 和 $\mathtt{sa}_i$ 分别属于 S, T，那么用 h_i 更新一下答案即可。
- 时间复杂度 $O(m \log m)$。
- 直接连接 S, T 也是可以的：为什么?

SP1812 LCS2 - Longest Common Substring II

题目描述

给定 n 个字符串 $s_1, \cdots, s_n$，求出 $s_1, \cdots, s_n$ 的最长公共子串的长度。
$1 \le n \le 10$，$\sum s_i \le 10^5$。

解法思路

- 连接 $s_1, \cdots, s_n$ 后做一遍 SA。
- 如果 $s_1, \cdots, s_n$ 都有一个后缀在 $\mathtt{sa}_l, \cdots, \mathtt{sa}_r$ 中出现了，那么就能用 $\min_{i=l+1}^{r} h_i$ 更新答案。
- 时间复杂度 $O(m \log m + mn)$。
- 如果 $n \le 10^5$ 呢？做到 $O(m \log m)$？

P1117 [NOI2016] 优秀的拆分

题目描述

如果一个字符串可以被拆分为 $AABB$ 的形式，那么把 (A, B) 称为其的一组优秀的拆分。

给定字符串 S，求出 S 的所有子串（位置不同算多次）的优秀的拆分个数之和。

$1 \le T \le 10$，$1 \le |S| \le 3 \times 10^4$。

解法思路

- 考虑求出 f_i, g_i 分别表示以 i 结尾的 AA 串的个数和以 i 开头的 BB 串的个数，那么答案就是

$$\sum_{i=1}^{n-1} f_i g_{i+1}$$

- 接下来考虑如何求 f_i，g_i 同理，枚举 A 的长度 k，考虑设置一些标兵：$k, 2k, \cdots$。
- 那么任何一个 AA 都会恰好覆盖相邻的两个标兵，枚举相邻的两个标兵，计算一下 LCP 和 LCS，就能知道哪些 AA 是合法的且覆盖了这两个标兵，差分修改即可。

P4022 [CTSC2012] 熟悉的文章

题目描述

给定一个 M 个 01 串，称一个串是熟悉的当且仅当这个串长度至少为 L 且在 M 个串中作为子串出现过。

现在给定一个长度为 N 的 01 串 A，询问最大的 L 使得，能够将 A 划分为若干子串，使得熟悉的子串的总长度不少于 $0.9N$，求出 L 的最大值。

输入文件总长度不超过 1.1×10^6 字节。

解法思路

- 预处理出 p_i 表示以 i 为右端点的最长的熟悉的串（不考虑长度至少为 L 的限制）。
- 接着 p_i 是单调的，直接二分 L 加上单调队列就能解决了。
- 问题就在于如何求解 p，可以使用 SA 或广义 SAM 均可。

CF700E Cool Slogans

题目描述

给定一个长度为 n 的小写字符串 w，你需要找到满足以下条件中长度最长的字符串序列 $s_1, \cdots, s_k$：

- 每个 s_i 都是 w 的非空子串；
- 对于 $i = 2, 3, \cdots, k$，要求 s_{i-1} 在 s_i 中至少出现了两次（可以重叠）。

求出最长的长度。

$1 \le n \le 2 \times 10^5$。

解法思路

- 发现可以让 s_{i-1} 是 s_i 的后缀，否则删除 s_i 多余的后缀也是合法的。
- 然后发现，这和 SAM 中 endpos 集合非常像。
- 更进一步地，假设我们知道了最后一个串，那么上一个串就是当前串不断在 fail 树上跳父亲直到第一个合适的。
- 反过来也是可以的，从根出发往下面走，碰到合法的就塞到末尾。
- 判断是否出现了两次的问题，只需要判断父亲的 endpos 集合中是否存在指定区间中的元素，可以使用主席树预处理出来。
- 时空复杂度 $O(n \log n)$。

24-25 集训队互测 Round 13 B. 字符串

题目描述

给定一个长度为 n 的字符串 $s[1:n]$。q 次询问，每次询问给定 i, r。求出有多少 l 满足如下条件:

- $1 \le l \le r$;
- $s[i:i+l-1]$ 字典序严格小于 $s[i+l:i+2l-1]$。

$1 \le n, q \le 5 \times 10^5$。

解法思路

- 考虑更加容易刻画的一个性质：$\texttt{rank}_i < \texttt{rank}_{i+l}$，这是充分的但不必要。
- 我们可以先计算满足这样的 l 的个数，再减去不符合条件的。
- 发现不符合条件的 l 需要满足：$s[i:i+l-1] = s[i+l:i+2l-1]$，且 $\texttt{rank}_i > \texttt{rank}_{i+l}$。
- 这和【优秀的拆分】一题很像，可以使用标兵法解决，但时间复杂度是 $O(n\log^2 n)$ 的。
- 要想做到 $O(n\log n)$ 需要使用 Runs。