

B 腊肠披萨

陈鸿基 / SpiritualKhorosho

Tsinghua University

May 26, 2024

简要题意

定义 $LCPS(s, t)$ 为最长的 r 使得 $s_1 \cdots s_{|r|} = t_{|t|-|r|+1} \cdots t_{|t|} = r$.
求

$$\sum_{i=1}^L \sum_{j=1}^L C^{|LCPS(s_i, s_j)|} \bmod P.$$

$$1 \leq L, \sum_{i=1}^L |s_i| \leq 3 \times 10^6, 2 \leq C < P < 2^{30}.$$

观察

虽然 *LCPS* 看上去没有什么规律, 但是注意到输入串的所有前缀和所有后缀的数量都是与总串长相等的 (如果去重则更少), 可以考虑用后缀去查是否存在对应的前缀, 并更新相应 (s_i, s_j) 的答案. 对前缀建立 Hash 表, 即可 $\Theta(1)$ 查询每个后缀.

这个做法有一个问题: 一个前缀可能对应多个原串, 如果需要统计每对 (s_i, s_j) 的 *LCPS* 长度再计算答案, 复杂度可能会爆炸.

因此, 我们需要设计一种方法, 通过查询到的前缀直接计算该前缀的贡献, 并且想办法容斥保证每对 (s_i, s_j) 只有匹配上的最长前缀/后缀的原始贡献计入答案.

容斥

考虑什么时候一个串 s_i 会有多个前缀被另一个串 s_j 的相应后缀匹配上. 假设已知 $s_{i,1} \cdots s_{i,k} = s_{j,|s_j|-k+1} \cdots s_{j,|s_j|}$ 且 $s_{i,1} \cdots s_{i,l} = s_{j,|s_j|-l+1} \cdots s_{j,|s_j|}$, 其中 $l < k$, 那么不难发现:

容斥

考虑什么时候一个串 s_i 会有多个前缀被另一个串 s_j 的相应后缀匹配上. 假设已知 $s_{i,1} \cdots s_{i,k} = s_{j,|s_j|-k+1} \cdots s_{j,|s_j|}$ 且

$s_{i,1} \cdots s_{i,l} = s_{j,|s_j|-l+1} \cdots s_{j,|s_j|}$, 其中 $l < k$, 那么不难发现:

$s_{i,1} \cdots s_{i,l}$ 应该是 $s_{i,1} \cdots s_{i,k}$ 的一个 border!

这启发我们可以对每个串跑 KMP, 用 fail 来处理容斥. 注意到 border 的 border 仍是 border, 这个容斥的形式非常好看.

一个前缀会在每个 border 处统计一次该 border 的贡献. 因此, 我们只需要将当前前缀的原始贡献减去其最长 border 的原始贡献.

把一个前缀的原始贡献拆分成其 border 贡献之和后, 我们就可以在查询后缀时直接把对应贡献乘上前缀出现次数计入答案即可.

总复杂度 $O\left(\sum_{i=1}^L |s_i|\right)$. 注意 C 的幂次可以递推, 如果每次暴力算会多个快速幂的 \log , 有可能无法通过本题.