

The background is a deep blue gradient with a subtle pattern of white dots. Overlaid on the left side are several concentric circles and arcs in a lighter blue color. Some of these arcs have degree markings, such as 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, and 260. There are also small white arrows pointing in various directions, suggesting a sense of rotation or movement.

字符串

煮僵尸 陈知轩

字符串的 BORDER

对于串 S ，设 $|S| = n$ 。

若 $S[1, i] = S[n - i + 1, i]$ ，则 $S[1, i]$ 是 S 的 Border。

设 $B(S)$ 为 S 的所有 Border（包括其本身），设 $LB(S)$ 为 S 除本身以外的最长 Border。

则：

1. $B(S) = B(LB(S)) \cup \{S\}$ 。
2. $B(S)$ 可以分成 $\log n$ 段等差数列（Border 引理）。

E.G.1 【模板】失配树

题目描述

给定串 $S, |S| = n$, 有 m 个查询, 每次询问给定 p, q , 查询:

$$\max\{ |t| \mid t \in \text{Border}(S[1:p]) \cap \text{Border}(S[1:q]) \}$$

$n, m \leq 10^6$ 。

E.G.1 【模板】失配树

简要思路

模板题当然不能写模板辣。

用 Border 引理写，每次选择 p, q 中较大的一者处理（设 $p > q$ ）：

设 $d = p - fail_p$ 。

1. 若 $d \geq \frac{p}{2}$ ，则直接 $p = fail_p$ 。
2. 则 $p = p \bmod d + d$ 。

注意当 $p - nxt_p$ 和 $q - nxt_q$ 相等时，它们可能在同一条链上，需要特判。

E.G.2 [WC2016] 论战捆竹竿

题目描述

给定串 $S, |S| = n$, 要求把 S 自身拼接若干次, 每次拼接可以共用相同的前缀和后缀。

如 aba 可以拼成 $aba, ababa, abaaba$ 。

求在 $[1, w]$ 中有多少种长度是可以通过拼接得到的。

$n \leq 10^6, w \leq 10^{18}$

E.G.2 [WC2016] 论战捆竹竿

简要思路

能拼出 d 就一定能拼出 $n + d$ ，因此问题可以转化为 $\text{mod } n$ 的同余最短路。

而每次就是加入一个 Border。

由于每一个点的出边都相同，这个特殊的最短路可以依次对于每一条考虑。

可以依次对于每一个等差数列进行转移，设其为 $x, x + d, \dots, x + l \cdot d$ 。

一开始 $\text{mod} = n$ ，每次加入一段等差数列时，可以将 mod 变为 x 。

把原序列按照 $+d$ 分成若干个环，每一个点 i 的出边的到达 $(i, i + l]$ 。

取 dis 最小的点作为环首，单调队列即可。

（实际上对于 $d > 1$ 的情况， $\sum x = O(n)$ ，所以暴力做也可以）。

（因为每次 x 减半）。

E.G.3 CTSC2006 歌唱王国

题目描述

给定一个串 S ，字符集为 $[1, m]$ 。

你有一个空串，现在你每次随机地向其末尾加入一个字符，当 S 成为该串子串时停止。

求期望加入了多少个字符后停止。

$|S| \leq 10^7, m \leq 10^9$ ，答案对任意数 P 取模。

E.G.3 CTSC2006 歌唱王国

简要思路

先扔答案：

$$\sum_{T \in B(S)} m^{|T|}$$

设 f_i 表示长度为 i 恰好结束的概率， g_i 表示长度为 i 仍未结束的概率。

$F(x), G(x)$ 分别为其 OGF，则答案就是 $\sum i \cdot f_i = F'(1) = \sum g_i = G(1)$ 。

假设在所有串后面加上一个 S 串，此时一定完成匹配，但可能并不是第一个匹配的位置。

而这样的情况说明原串包含了 S 的一个 Border。

枚举这样的 Border，得到：

$$G(x) \cdot \left(\frac{1}{m}x\right)^{|S|} = F(x) \sum_{T \in B(S)} \left(\frac{1}{m}\right)^{|S|-|T|}$$

带入 $x = 1, F(1) = 1$ ，得到 $G(1) = \sum_{T \in B(S)} m^{|T|}$ 。

E.G.4 [HDU-6791] 2020HDU多校第三场T1

题目描述

给定一个串 S ，和 m 个查询，字符集为大写字母。

每次查询它的两个 回文子串，比较这两个子串在上一个问题中的答案大小。

$|S|, m \leq 5 \cdot 10^5$ 。

E.G.4 [HDU-6791] 2020HDU多校第三场T1

简要思路

回文串的 `Border` 也是回文串，且回文自动机能处理这样的结构。

回文自动机的 `fail` 指向每一个回文子串的 `Border`。

只需要在 `fail` 树上比较字典序，可以 Hash 处理，也可以用 `Border` 的等差数列性质。

E.G.5 ?

题目描述

给定一个串 S 和一棵 *trie* 树，字符集为小写字母。

有 m 个询问，每次询问 $S[l:r]$ 在 *trie* 树上最深的匹配节点。

$|S|, m, |trie| \leq 10^6$ 。

E.G.5 ?

简要思路1

把 trie 树上每一个点的 Hash 值扔进 Hash Table 里。

查询时二分答案，然后在 Hash Table 里找，复杂度为 $O(\log n)$ 。

实现简单，但是不一定能过。

E.G.5 ?

简要思路2

给 trie 树剖，为重链维护 Hash 值。

在重链上二分+Hash，轻链最多跳 \log 次，复杂度为 $O(n \log^2 n)$ 。

换成全局平衡二叉树即可 $O(n \log n)$ ，常数较小。

E.G.6 「ROI 2016 DAY2」 二指禅

题目描述

给定一个串 S 和若干个模板串 T_i ，每一个模板串有权值 w_i 。

一开始你有一个空串。

你每次可以选择一个串 T_i ，花费 w_i 的代价将 T_i 的一段前缀或者一段后缀加入。

求最小的代价构造 S 或者判断不存在方案。

$$|S|, \sum |T_i| \leq 10^5$$



E.G.6 「ROI 2016 DAY2」二指禅

简要思路1

建立正反两棵 *trie* 树，用上一题的方法求找到最长匹配。

每一个节点最多有 \sqrt{n} 个祖先有对应的串，枚举所有的串进行转移。

转移的每一次都是一个：

1. 区间修改/单点查询
2. 单点修改/区间查询

且操作次数均为 $1:\sqrt{n}$ ，可以用分块处理，复杂度为 $O(n\sqrt{n})$ 。

E.G.6 「ROI 2016 DAY2」二指禅

简要思路2

建立压缩 trie ，在匹配时最多经过 \sqrt{n} 个关键节点。

剩下的部分相同。



E.G.7 CF1483F - EXAM

题目大意

给定 n 个不同的串 s_i ，设 $m = \sum |s_i|$ ，记 $s_i \subset s_j$ 表示 s_i 是 s_j 的子串。

求二元组 i, j 的个数，使得：

1. $i \neq j, s_i \subset s_j$ 。
2. 不存在 k 满足 $k \neq i, k \neq j, s_i \subset s_k \subset s_j$ 。

$n, m \leq 10^6$ 。

E.G.7 CF1483F - EXAM

简要思路

答案是 $O(n)$ 级别的：

对于每一个串 s_j 枚举其中 s_i 出现的右端点，对于特定的右端点只有最长的 s_i 可能合法。

用 AC 自动机即可找到这样的 s_i 以及其对应的区间 $[l, r]$ ，注意这样的区间可能有多个。

此时若对于 s_i 对应的某一个区间被另一个串 s_k 对应的区间包含则 s_i 不合法。

E.G.8 CF1393E2 - TWILIGHT AND ANCIENT SCROLL (HARDER VERSION)

题目描述

给定 n 个串 S_i 。

在每一个串中删除至多 1 个字符，使得 S_i 字典序不减。

求方案数，对于任意数 P 取模。

$\sum |S_i| \leq 3 \cdot 10^6$ 。

E.G.8 CF1393E2 - TWILIGHT AND ANCIENT SCROLL (HARDER VERSION)

简要思路

设 dp_i 表示上一个串 s 删除了第 i 个字符时的方案数。

枚举这个串 t 删了第 j 字符，找到两串第一个不一样的位置 k ，进行比较：

1. 若 $s_k < t_k$ ，则 $i > k$ 的情况都合法。
2. 若 $s_p \sim s_k$ 都与 s_k 相同，删除 $i \in [p, k]$ 会得到同一个 s' ，暴力比较 s', t' 即可。
3. 若 $i < p$ ，则删除导致的平移一定会使得 s', t' 在 k 之前就出现不同，且此时就是：
比较 s_i 与 s_i 后面第一个不同的字符。

在删除时最多产生 ± 1 的偏移，可以线性处理这样的 LCP 问题。

E.G.9 CF1276F - ASTERISK SUBSTRINGS

题目大意

给定一个串 S , $|S| = n$, 仅包含小写字母。

设 S'_i 表示把 S 的第 i 个字符替换为 $*$ 号的字符串。

求 $\{S, S'_1, S'_2, \dots\}$ 一共有多少本质不同的子串。

$n \leq 10^6$ 。

E.G.9 CF1276F - ASTERISK SUBSTRINGS

简要思路

不包含 $*$ 号的串比较简单。

否则对于一个后缀，删去后面若干字符得到的部分可以表示为后缀自动机上的一段祖先链（附带一个长度作为参数）。

建立正反两个后缀自动机，问题就变为了：

有若干个二元组 (a_i, b_i) ，求有多少个二元组 (s, t) ，

使得存在一对 a_i, b_i 满足 s, t 分别是 a_i, b_i 的祖先。

在第一棵树上用线段树合并维护 b ，容易发现这个问题同 [ZJOI2019]语言。

复杂度为 $O(n \log n)$ 。

E.G.10 ICPC 2021 沈阳 M STRING PROBLEM

题目大意

给定一个串 S ，对于其每一个前缀，找到字典序最大的一个子串。

$$|S| \leq 10^6$$

上述四种做法都是支持在线在结尾添加字符的，不过我们也允许使用离线算法。赛中通过这个题的队伍可谓是八仙过海各显神通，有用后缀数据结构（后缀树、后缀数组、后缀自动机）做的、也有用border性质做的、还有用lyndon分解做的，涵盖了多个时代的字符串科技。这些做法这里就不一

E.G.10 ICPC 2021 沈阳 M STRING PROBLEM

一种做法

维护后缀自动机，答案一定是在转移上每次选择最大的一个进行。
在插入字符时也动态修改这条答案对应的转移链即可。



E.G.10 ICPC 2021 沈阳 M STRING PROBLEM

另一种做法

推论：

1. 任意时刻答案都是当前前缀的某一个极长后缀。
2. 每次增加一个字符时，新的答案都是当前答案的一个 **Border** 再加上新的字符。

E.G.11 ??

题目大意

给定 n 个串 S_i 和 m 个查询。

对于每一个查询，给你一个字符串 T ，询问有多少个对 (i, j) 满足 T 可以表示为 S_i 的后缀与 S_j 的前缀拼接的结果（ i, j 可能相等）。

$\sum |S_i|, \sum |T| \leq 10^6$ ，要求空间线性，时间可以稍微放缓一点。

E.G.11 ??

一种做法

将 S_i 的正串反串分别拿出来建 trie 树。

枚举两部分的断点，分别在两棵 trie 树上找到 $T[1:k], T[k+1:]$ 对应的节点，那么合法的 S_i, S_j 分别在其子树中。

子树里的点对应一段编号的区间，问题转化为 $[a,b]:[c,d]$ 是合法的，也就是矩形并的面积。可以扫描线。

复杂度为 $O(n \log n \sim n \log^2 n)$ 。

std 好像是 kmp 自动机。

E.G.12 CTSC 2016 香山的树

题目大意

对于所有长度 $\leq n$ ，且由小写字母构成的串 S 。

若 S 是其本身最小的循环同构，且这样的同构位置唯一，则称 S 是好的。

你要维护：

1. 查询一个串在所有好的串中的字典序排名。
2. 查询好的串中字典序第 k 大的串。

$n \leq 50, k \leq 10^{18}$ 。

E.G.12 CTSC 2016 香山的树

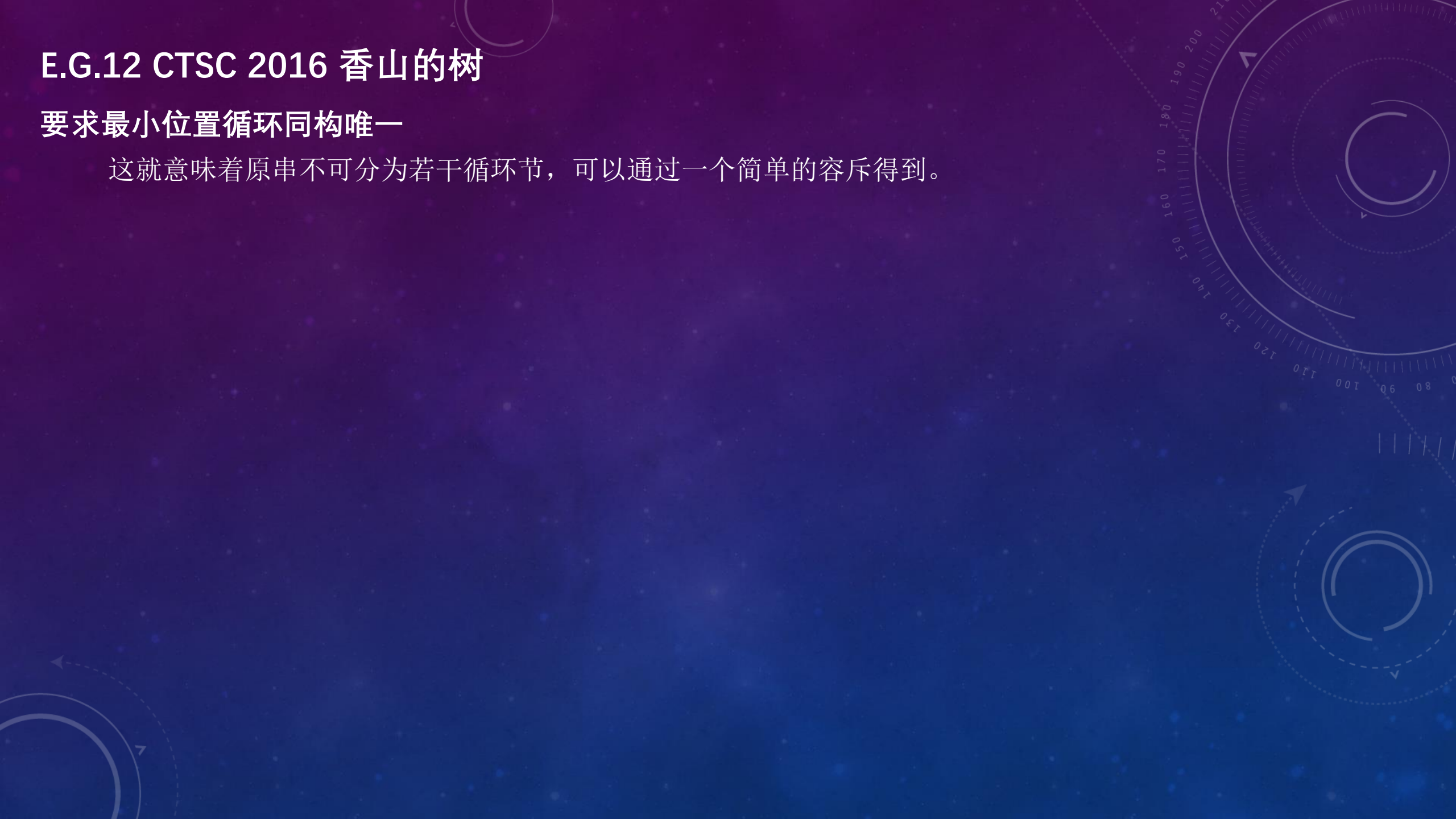
简要思路

实际上本质还是要计算字典序 $\leq S$ 的好串有多少个，也可以算 $> S$ 的个数。
由于 n 非常小，可以随意枚举答案的每一位， dp 只要不是太离谱都能过。

E.G.12 CTSC 2016 香山的树

要求最小位置循环同构唯一

这就意味着原串不可分为若干循环节，可以通过一个简单的容斥得到。



E.G.12 CTSC 2016 香山的树

比较字典序

考虑计算 $> S$ 的方案数，则最小字典序出现的位置与原串 S 有最大匹配。

考虑 kmp，设最优匹配长度为 l ，一个循环节一定从 l 开始回到 l 。

枚举 l ，令 $dp_{i,j,k}$ 表示长度为 i ，当前匹配为 j ， l 出现了 k 次。

1. 一开始 $j = l$ ，枚举每一位的字符进行转移。
2. 每次出现失配时，新增的字符一定比失配位置的字符大。

预处理 kmp 自动机得到失配位置。

3. 最后对于 k 进行倍数容斥。