字符串简单题选讲 1

Purslane

云斗学院 · 省选培训 · 第一阶段

2025.12.22

记号与约定

给定字符集 Σ，定义字符串 S 为 Σ 中的一些字符有序拼接而成的序列。记 $n = |S|$ 为字符串长度；$\forall 1 \le i \le n$，记 S_i 表示字符串从左往右第 i 个字符。

记 $S[l \cdots r]$ 为 $S_l S_{l+1} \cdots S_r$，是字符串的一个子串（注意区分子串与子序列的区别）。

有部分字符串问题，要求字符集 Σ 的大小为常数（如，小写字母集）；而有些不需要。学习的时候应该加以区分。

为什么要学习字符串问题

字符串问题一个很突出的特点是：你没学过，大概做不出来。
为了避免被字符串问题创飞（如 NOI2023，CSP2025 等），日常的学习中不能忽视字符串的训练。
今天我们主要介绍：字符串匹配相关，字符串周期相关。
明天我们将介绍：后缀结构相关，回文串相关。
我们将不介绍后缀自动机（10 级知识点），感兴趣的同学可以自行查阅资料，

字符串匹配

字符串 S 与字符串 T 相同，当且仅当他们长度相同并且对应位置的字符串相同。
有的时候，我们希望知道 T 与 S 中哪些子串相同，称这样的问题为字符串匹配问题，S 为文本串，T 为模式串。

(1) 字符串哈希；
(2) KMP 算法（可以解决单模式串问题）；
(3) AC 自动机（可以解决多模式串问题）。

字符串哈希

基础思想：将字符串视作多项式，将字符串相等视作多项式相同。
对于字符串 S，定义 $f_S(x) = \sum_{i=1}^{n} \phi(S_i)x^i$，其中 ϕ 是 $\Sigma \to \mathbb{C}$ 的一个单射（一般而言使用 $1 \sim |\Sigma|$）。
对于字符串 S 和 T，记 $H_{S,T}(x) = f_S(x) - f_T(x)$。若 $S = T$，则 $H_{S,T} \equiv 0$；否则。当 $0 \neq H_{S,T}(x) \in \mathbb{Z}[x]$ 时，$H_{s,t}(x) \equiv 0 \pmod P$ 至多 $\max\{|S|, |T|\}$ 个根。
因此，随机选取 $b \in \mathbb{Z}$，有 $1 - \frac{n}{P}$ 的概率可以通过函数值来区分 $f_S(x)$ 和 $f_T(x)$。
然而根据生日悖论，当 $P \approx 10^9$ 时很容易找到 $S_{1,2} \in S$ 使得 $S_1 \neq S_2$ 而 $f_{S_1}(b) = f_{S_2}(b)$（这不是基于模数定向 hack 的）。
一种可能的方法是：双模数哈希。

为什么哈希是神

字符串哈希可以以更高的复杂度（大部分情况只是多了个 $\log$）去替换很多复杂的字符串算法。
如 $O(n\log^2 n)$ 进行后缀排序（以及 $O(\log n)$ 计算两个串，可以是后缀等，的 LCP），$O(n\log n)$ 实现 Manacher 的功能。
接下来，再介绍字符串问题之前，我们额外介绍两个哈希在数据结构中的运用。

P12736 [POI 2016 R2] 圣诞灯链 Christmas chain

题目大意

有 n 个灯泡，编号分别为 $1, 2, \cdots, n$。有 m 条限制，限制 (a, b, t) 表示，$\forall 1 \le i \le t$，第 $(a+i-1)$ 和第 $(b+i-1)$ 个彩灯颜色相同。

问最多能有多少种颜色。

$1 \le n, m \le 5 \times 10^5$，4.5s。

P12736 [POI 2016 R2] 圣诞灯链 Christmas chain

本题存在很聪明的 $O(n\log n\alpha(n))$ 做法，这里介绍 $O(n\log^2 n+m\log n)$ 的暴力解法。

很直接的想法是：直接使用并查集。然而，我们有 $O(nm)$ 次“合并操作”，不可能直接套用。注意到本质上只有 $O(n)$ 次有效的合并，我们把他们都找到即可。

记 dsu_u 表示第 u 个彩灯当前所属的并查集编号。我们将执行以下流程：

(1) 找到最小的 $i\geq 1$，使得 $dsu_{a+i-1}\neq dsu_{b+i-1}$。

(2) 如果 $i>t$，退出；

(3) 否则，合并 $a+i-1$ 与 $b+i-1$，重新构建 dsu 序列。

容易发现，第一步和计算 LCP 本质相同。

使用线段树 + 二分 + 启发式合并，容易做到 $O(n\log^2 n+m\log n)$。

循环背包

题目大意

给定 n 个正整数 $a_{1,2,\cdots,n}$，以及模数 P。
对每个 $0 \le i < P$，判断是否能从 a 中选择若干个数，使得他们的和模 P 余 i。
$n, P, a_i \le 10^6$。

循环背包

处理方法与上一道题类似：我们记 dp_i 表示当前 i 能否被表示出来。每一轮，如果 $dp_{i-a}=1$ 而 $dp_i=0$，则可将 dp_i 变为 1（是 01 背包而不是完全背包，注意顺序）。

找到所有的 i 使得 $dp_i \neq dp_{i+a}$（使用哈希 + 二分实现）。假设有 t 个这样的 i，可以证明：我们恰好会将 $t/2$ 个 dp 的值从 0 变成 1。

因此，均摊复杂度为 $O(P\log^2 P + n\log P)$。

P3823 [NOI2017] 蚯蚓排队

题目大意

有 n 个字符串，每个字符串初始时只有一个字符。
进行以下三种操作：

(1) 将两个字符串拼接在一起（原有的两个字符串将删除）；

(2) 将一个字符串从中间分裂成两个字符串，该操作最多进行 c 次；

(3) 给定字符串 T 和正整数 k，查询 T 的每个长度为 k 的子串在当前所有字符串出现次数之和。

$1 \le n \le 2 \times 10^5$，$1 \le c \le 1000$，$1 \le m \le 5 \times 10^5$，$1 \le k \le 50$，$\sum |T| \le 10^7$，3s。

P3823 [NOI2017] 蚯蚓排队

一个很暴力的想法是，开一个 umap 之类的东西，维护当前所有长度 ≤ 50 的子串出现次数之和（子串当然使用哈希来维护）。
当将两个字符串拼在一起或者分开的时候，枚举所有长度 $\leq k$ 且跨过分界线的子串。
处理分裂的复杂度显然为 $O(ck^2)$ 可以接受；处理合并的复杂度为 $O(nk + ck^2)$！（很容易分析为 $O(nk^2)$。实际上，记 lst_u 表示每个字符 u 之后又多少个字符，容易通过对 lst 的势能分析得出 $O(nk + ck^2)$ 这个复杂度）
查询复杂度为 $O(|T|)$。
代码可能有点 Dirty Work，以及卡常。

P3449 [POI 2006] PAL-Palindromes

题目大意

定义 $f(S) = [S\text{是回文串}]$。给定字符串 $S_{1,2,\cdots,n}$，求 $\sum_{i=1}^{n}\sum_{j=1}^{n} f(S_i + S_j)$，其中 $+$ 表示字符串拼接。$\sum |S_i| \le 2 \times 10^6$，1s。

P3449 [POI 2006] PAL-Palindromes

容易特判 $|S_i| = |S_j|$ 的情况，不妨设 $x = |S_i| < |S_j| = y$。
那么只需要：

(1) $S_j[1 \cdots (y-x)]$ 是回文串；

(2) S_i 与 $S_j[y \cdots (y-x+1)]$ 相同（注意，这是反向的字符串）。

预处理 j、枚举 i，容易使用哈希 + umap 实现。

其他可能的哈希类型

不同的组合对象，可能有截然不同的哈希方法（上文中，拍成多项式是对于这种线性序列的哈希方法）。
感兴趣的同学可以自行查阅资料，下面给出一些（文字是可以点开的链接）：

(1) 循环同构哈希。
(2) 树哈希。
(3) 集合哈希（随机赋权方法）。

一份不错的哈希题单。

Border 的概念

定义字符串 S 的 Border 为：S 真子串 T，使得 T 同时是 S 的前缀和后缀。

如果 $|T_1| < |T_2|$ 满足 $T_{1,2}$ 均为 S 的 Border，那么 T_1 为 T_2 的 Border。

由此，记 $b(S)$ 为 S 的最长 Border，有：$b(S), b(b(S)), \cdots$ 恰好构建了 S 的 Border 集合。

容易知道，$S[1 \cdots i]$ 的 Border 也应该是 S 的一个前缀 $S[1 \cdots j]$（$j < i$，特别的 $j = 0$ 可能成立）。

因此，记 $fa_i = |b(S[1 \cdots i)]$，那么 $i \to fa_i$ 连边得到了一个树的结构（称之为失配树）；$S[1 \cdots i]$ 的所有 Border 就是 $i \to 0$ 路径上的所有节点。

问题：给定字符串 S 和 T。已知 i_0 是最大的数使 $S[1 \cdots i_0]$ 是 T 的后缀。那么，所有使得 $S[1 \cdots i]$ 是 T 后缀的 i 如何刻画？

KMP 算法

回应上一页 PPT 的问题，显然这样的 i 是失配树中 $i_0 \to 0$ 路径上的节点。

因此，我们很容易得到 KMP 算法的流程：如果要分析模式串 T 在文本串 S 中的出现情况，那么先建立出 T 的失配树；然后从左往右扫描 S 中的字符，我们维护是 S 后缀的最长 T 前缀。

KMP 算法充分利用了 T 串自身结构的性质，做到了均摊 $O(n+m)$。

注意到，我们在匹配的过程中不需要知道 S 之前的结构，只需要知道当前匹配到 T 的哪个位置；假设当前在 T 的位置 u，加上字符 c 之后到了 $f(u,c)$，那么我们得到了一个自动机——KMP 自动机。

容易发现，KMP 自动机本质上就是单串的 AC 自动机，因此 KMP 自动机上 DP 大多比较无聊，不如 AC 自动机上 DP。

P3546 [POI 2012] PRE-Prefixuffix

题目大意

给定一个字符串 S。求最大的 $2L \le |S|$ 使得 S 长度为 L 的前缀与后缀是循环同构的。

$|S| \le 10^6$，1s。

P3546 [POI 2012] PRE-Prefixuffix

显然 S 形如 $AB\cdots BA$ 的时候（B 可以为空串）$L=|AB|$。
我们可以枚举 A，问题变为：计算 $S[i,n-i+1]$ 不重叠的最大 Border。
容易发现，$S[i,n-i+1]$ 的 Border 可以对应到 $S[i+1,n-i]$ 的 Border 上去，因此计算这个事情也是均摊 $O(n)$ 的。

P4548 [CTSC2006] 歌唱王国

题目大意

给定序列 $a_{1,2,\cdots,n}$ $(1 \le a_i \le m)$。我们将在 $1 \sim m$ 范围内等概率随机生成序列 b。

求 a 在 b 中第一次出现的位置的右端点的期望值（可以证明，是整数）对 10000 取模。

TestCase $= 50$，$1 \le n \le 10^5$，3s。

P4548 [CTSC2006] 歌唱王国

容易发现，本质上是建出 KMP 自动机后，进行随机游走（从 0 点开始），问走到终点的期望时间。
注意 KMP 自动机的结构很特殊：每个节点只有一条出边连向编号比他大的节点，其他边均连向编号比他小的节点。
因此可以套用P6835 [Cnoi2020] 线形生物的方法。
问题在于，本题的字符集很大，不能直接枚举出边。对 dp 数组做前缀和后得到 pre 数组，我们需要关系 $del_i = \sum_{c \neq a_{i+1}} pre_{f(i,c)}$，其中 f 是 KMP 自动机的转移函数。
容易发现，del_i 和 del_{nxt_i} 只有 $O(1)$ 的差别，暴力做就行，复杂度 $O(n)$。（预处理需要一遍 DFS）

字典树

即 Trie 树，用来维护给定的字符串集合（字典）的前缀信息。
Trie 树有很多神神秘秘的小技巧，比如贪心、全局 $+1$，以及把 Trie 树视作普通的树在上面做树上问题（如 DSU On Tree，树链剖分等）。
鉴于这些东西和“字符串”的关系不大，我们就不详细介绍了。
一些例题：

(1) P11160 【MX-X6-T6】機械生命体。（从低位往高位建，Trie 树合并，Trie 树打 Tag）
(2) P10218 [省选联考 2024] 魔法手杖。（Trie 树上贪心）
(3) P4592 [TJOI2018] 异或。（Trie 树 + 简单树上问题）

AC 自动机

AC 自动机旨在处理多模式串匹配问题假设模式串集合为 D，建立出他们的字典树；文本串 S 的、能对应到字典树上某一个节点的最长后缀长度为 L。

其他能在字典树上对应到某些东西的后缀，也应该具有 Border 结构，我们称之为 Fail。在字典树上预处理出 $fail_u$ 表示 $root \to u$ 对应的字符串，在字典树能对应的最长真后缀（使用 BFS 建立）。那么我们得到新的 Fail 树（有别于字典树），可以很方便的通过树上算法处理这些问题。

我们仍然希望得到“假设当前在节点 u，S 的下一个字符加入 c 的时候，会转移到哪个节点去”，这种自动机结构称为 AC 自动机。

很容易 $O(|\Sigma| \sum_{T \in D} |T|)$ 建立字典图，对于字符集更大的情况可以使用主席树——在 Fail 树上 BFS，然后可持久化修改。

P5840 [COCI 2014/2015] Divljak

题目大意

有一个字符串集合 T，初始为空集；还有 n 个字符串 $S_{1,2,\cdots,n}$。进行以下两种操作：

(1) 向集合 T 中插入字符串 P；

(2) 问 T 中有多少个字符串包含 S_x。

T 中字符串长度总和 $\leq 2\times 10^6$，$\sum |S_i| \leq 2\times 10^6$，4s。

P5840 [COCI 2014/2015] Divljak

显然，要建立出 S 的 AC 自动机。那么插入一个 T，假设 T 在 AC 自动机上经过的节点集合为 T_0，那么 T_0 中节点到 $root$ 的路径并上所有节点的答案加一。

如何处理这样的路径并？只需要按照 DFS 序排序，就变成了树上单点加、子树查询问题。

复杂度 $O(L(\log L + |\Sigma|))$，其中 L 表示字符串总长度。

QOJ6842 Popcount Words

题目大意

有一个无穷长的字符串 S，$S[i] = \text{popcount}(i) \bmod 2$。
给定 n 个区间 $[l_i, r_i]$，字符串 $T = \sum_{i=1}^{n} S[l_i, r_i]$。
进行 q 次询问，每次给定一个字符串 Q，问 Q 在 T 中出现了多少次。
$\sum |Q| \le 5 \times 10^5$，1s。

QOJ6842 Popcount Words

S 显然具有很好的对称性，我们希望充分利用这个性质。
首先，对询问串建出 AC 自动机，我们先尝试在上面模拟游走。
容易想到，将 T 拆成若干个形如 $[a \times 2^t, (a+1) \times 2^t)$ 的区间，每个区间里 popcount 函数是相当有规律的，并且只和二元组 $(t, \text{popcount}(a) \bmod 2)$ 有关。
对每个节点，预处理 $nxt_{u,t,0/1}$ 表示，从 u 开始，走 2^t 步，有额外 0 或 1 的增量。
容易发现，$nxt_{u,t,o} = nxt_{nxt_{u,t-1,o},t-1,o\oplus 1}$。
然后考虑计算每个点在往自动机里放 T 的过程中被经过了多少次，那么记 $cov_{u,t,o}$ 为每个点作为 (t,o) 二元组的起点经过了多少次，将贡献摊到 $cov_{u,t-1,o}$ 和 $cov_{nxt_{u,t-1,o},t-1,o\oplus 1}$ 上即可。
复杂度 $O((\sum |Q|)\log V)$。

CF1483F Exam

题目大意

给定 n 个互不相同的字符串 $S_{1,2,\cdots,n}$。
(i,j) 之间会发生一场对决，当且仅当：

(1) $i \neq j$；

(2) S_j 是 S_i 的子串；

(3) 不存在 $k \neq i,j$ 使得：S_k 是 S_i 的子串，S_j 是 S_k 的子串。

问一共有多少场对决。
$\sum |S_i| \leq 10^6$，4s。

CF1483F Exam

给定 S_i，计算哪些 S_j 满足 S_j 是 S_i 的子串。这个就是 P5840，结构就是 Fail 树上若干根到路径的并。
如果存在 $S_j \subset S_k \subset S_i$，那么这样的 j 也应该是 Fail 树上若干到根的路径的并。
考虑加入 S_j 在 S_i 的位置 p 出现了（末尾）。那么如果存在 $p \le q$ 使得：S_i 以 q 为末尾的最长串（不包括 $p = q$ 时的 S_j），左端点覆盖了 $p - |S_j| + 1$，那么 S_j 就被击杀了。
所以，我们也可以 $O(|S_i| \log n)$ 的计算出不合法的结构。总复杂度 $O((\sum |S_i|) \log n)$。

CF1801G A task for substrings

题目大意

给定 n 个字符串构成的集合 S，以及字符串 T。
定义 $f(T)$ 为，S 中所有元素在 T 中出现次数之和。
有 m 次询问，每次给出 $1 \le l \le r \le |T|$，求 $f(T[l \cdots r])$。
$\sum_{s \in S} |s| \le 10^6$，$|T| \le 5 \times 10^6$，$m \le 5 \times 10^5$，4s。

CF1801G A task for substrings

定义 $g(T) = [T \in S]$。记 a_i 为，$\sum_{j=1}^{i} g(T[j \cdots i])$，这是 AC 自动机的经典问题。
我们想要把 $\sum_{i=l}^{r} a_i$ 当做答案，不过这显然有问题——可能会把 $L < l \le R \le r$ 的 (L, R) 算计答案中去。
记 i_0 为最大的 i 使得，a_{i_0} 中统计了左端点在 l 之前的串。
我们只需要计算 $f(T[l \cdots i_0]) + \sum_{i=i_0+1}^{r} a_i$。
而 $T[l \cdots i_0]$ 一定是 S 中某个串的后缀！建立反串的 AC 自动机，即可轻松解决。

AC 自动机的其他方法

容易发现，AC 自动机不支持动态增加新的字符串，可能的方法是二进制分组。
参考资料：CF710F String Set Queries，带删除的二进制分组。
推荐一道“自动机”（和 AC 自动机无关）的问题：CF506E Mr. Kitayuta's Gift。

字符串匹配问题的其他方法

看看就行，在考试中 impractical。

(1) 使用多项式方法，处理带通配符的字符串匹配等问题，参考资料，QOJ5107 Mosaic Browsing。

(2) 使用 bitset 乱搞字符串匹配，可以处理一些很复杂的字符串问题，P4465 [国家集训队] JZPSTR、参考资料。

字符串的周期

给定长为 n 的字符串 S。
对于 $p < n$，如果 $\forall i > p$ 都有 $S[i] = S[i-p]$，称 p 是 S 的一个周期。
容易发现，p 是 S 的周期，等价于 S 有一个长度为 $n-p$ 的 Border。所以 S 的最小周期就是 n 减去 S 的最大 Border 长度。
我们将通过分析字符串的周期，得到字符串 Border 的一些有趣的小性质。

周期引理

如果大家学过高中数学，大概可以知道：若函数 $f(x)$ 的最小正周期 T 存在，那么任何一个周期 t 均为 T 的正整数倍。
这涉及到一个核心结论：如果 $u \neq v$ 均为 f 的周期，则 $|u-v|$ 也为 f 的周期。
在字符串问题中，我们也有结论：若 (p,q) 是 S 的周期，那么当以下两个条件满足一个的时候，$\gcd(p,q)$ 也是 S 的周期：

(1) $p+q \leq |S|$；

(2) $p+q-\gcd(p,q) \leq |S|$。

前者被称为弱周期引理，后者被称为周期引理。
因此，记 $p = |S| - \text{maxborder}(S)$，那么如果 S 有长度为 t 的 Border，满足 $p \leq t$，则 $p \mid |S|-t$。
这也可以很自然的推出一个结论：S 所有长度超过 $\frac{|S|}{2}$ 的 Border 长度，构成等差数列。

Border 的一些其他性质

容易证明，一个字符串的 Border 可以划分为 $O(\log n)$ 个等差数列。我们可以把它直接构造出来。

一个比较容易的构造方法是，取长度在 $[2^k, 2^{k+1})$ 中的 Border，他们显然构成等差数列，所以等差数列的个数可以严格控制到 $\lceil \log_2 |S| \rceil$。

有一个有趣的小结论：如果字符串 S 和 T 满足 $2|S| \geq |T|$，那么 S 在 T 中出现的位置是等差数列。

根据这个性质，我们可以尝试解决区间 Border 问题：给定字符串 S，有若干次询问，每次询问求出 S 某个子串的所有 Border（用等差数列的形式给出）。

这个问题叫做基本子串字典，如果有时间我们可以讲一下它的核心思路，细节可以参考这篇文章。

整周期

如果字符串 S 可以被划分为若干个长度为 d 的完全相同的子串，那么 S 就有整周期 d（不妨设 $d < |S|$）。
那么显然，S 的最小正周期应该是 d 的因子，因此只需要求出 Border 就可以弄明白整周期是否存在，以及是哪些。

P9717 [EC Final 2022] Binary String

题目大意

给你一个 01 字符串 S。每一轮，找到所有的 $(i, i+1)$ 使得 $S[i] = 0$ 且 $S[i+1] = 1$，同时将这样的 $(S[i], S[i+1])$ 替换为 $(1, 0)$。($i = n$ 时定义 $i + 1 = 1$)

问在充分长时间后，会出现多少个本质不同的字符串，对 998244353 取模。

$|S| \le 10^7$。2s。

P9717 [EC Final 2022] Binary String

不妨设序列中 1 的个数不少于 0 的个数。在若干轮之后，所有 0 连续段的长度都会变为 1。
此时每一轮相当于对字符串做一次循环位移，这一部分的问题变为：给定字符串 S，问它的循环同构本质有多少种可能。容易发现就是整周期数，可以 KMP 求。
现在考虑怎么求出这个“终止状态”。观察所有长度 ≥ 2 的 0 连续段和 1 连续段，发现他们的变化是：

(1) 每个时刻，0 连续段尝试往左平移一段，1 连续段尝试往右平移一段；

(2) 如果一个 0 和一个 1 撞上了，这两段的长度都会各自减去 1（如果减成 1 了就将其删掉）。

P9717 [EC Final 2022] Binary String

这个过程很像拿 1（左括号）去“括号匹配”掉 0（右括号）。
希望使用栈来实现。然后本题是环状的，我们不希望末尾的 1 去匹配最前面的 0。
一种可能的方法是使用 Raney 引理，将环进行位移后断环成链，这样就可以变为序列问题。
复杂度 $O(n)$，实现起来可能有一点 Dirty Work 和 Corner Case。
Raney 引理：给定序列 a，如果 $\sum a \geq 0$，那么存在 a 的一个循环位移，使得 $\forall 1 \leq i < n$，$\sum_{j=1}^{i} a_j > 0$。

P1393 Mivik 的标题

题目大意

有一个长度为 t 的字符串 S，每个字符为 1 到 m 之间的数字。问随机生成一个长度为 n 的字符串，包含 S 作为子串的概率为多少。

$|S| \le 10^5$，$n \le 2 \times 10^5$，$m \le 10^8$，1s。

P1393 Mivik 的标题

考虑使用容斥原理，钦定若干位置往后延伸 t 位恰好为 S，那么需要考虑相邻两个钦定的位置 p_1 和 p_2 是否满足要求。

(1) 如果 $p_1 + t \le p_2$，那么显然满足，并且新增的概率为 m^{-t}；

(2) 否则，需要 $t - p_2 + p_1$ 是 S 的一个 Border，新增的概率为 $m^{-p_2+p_1}$。

对于前者，只需要前缀和优化。对于后者，注意到给定 p_2 时，p_1 可以划分为 $O(\log n)$ 个等差数列，且这 $O(\log n)$ 个等差数列的公差与 p_2 无关（只和 S 有关）。
维护 $O(\log n)$ 个前缀和即可，复杂度 $O(n \log n)$。
本题存在一些多项式方法，感兴趣的同学可以自己研究。

P5287 [HNOI2019] JOJO

题目大意

字符串 S 初始为空集，我们进行 n 次以下两个操作之一：

(1) 往 S 末尾增加 t 个字符 c。方便起见，保证操作前 S 末尾字符不是 c。

(2) 回退到某个版本。

每次操作之后，求出所有前缀的最长 Border 之和，对 998244353 取模。

$n \le 10^5$，$x \le 10^4$，2s。

P5287 [HNOI2019] JOJO

首先先建操作树，可持久化是假的。
思路 1：直接模拟 KMP 自动机的过程。
我们在 $p = border(S_0)$ 的基础上，往后跳 t 次 c。
分类讨论 p 的位置：

(1) 如果 p 不是某一段的末尾，那么再跳一次 c 显然只需要 $p \leftarrow p + 1$；
(2) 如果 p 是某一段的末尾，那么此时要跳 border。继续分类讨论：
(3) 如果新的 border 和这个 p 在同一段中，那么之后再跳就会陷入一个循环，是容易处理的；否则，重新回到 (1)。

如果颜色段 u 末尾的 border 指向颜色段 v，那么 $u \to v$ 连边。
那么我们在这棵树上跳父亲（树上每个节点内部可跳 t 次）。
相当于给你一棵树，你要从某个点不断跳父亲，使得经过的点的权值 $\geq v$ 就停下。使用倍增容易处理，复杂度 $O(n \log n)$。

P5287 [HNOI2019] JOJO

思路 2：考虑压缩字符。
考虑一个 Border，掐头去尾后的若干段，应该是完全相同的；首尾满足一些简单的不等式就行。因此考虑从第二个颜色段开始，我们计算这里面的 Border。
这时候不断跳 Border 的复杂度均摊是错的，很伤心。不过没关系！Border 的等差数列性质，蕴含着周期——实际上跳 Border 只需要 $O(\log V)$。
总复杂度为 $O(n \log V)$，可以接受。

n 次方串

如果一个字符串可以写成 n 个 A 拼接而成，称之为一个 n 次方串。
记 $|A| = L$，如果 $|A|$ 的最小正周期 $p \mid L$，那么 $|A|$ 是 d 次方串成立，等价于 $d \mid \frac{L}{p}$。
否则，$|A|$ 最多只能是 1 次方串。
Runs 是处理 n 次方串的有利武器，但由于 Runs 实际上并不 practical，因此只提供参考资料。

QOJ15502 字符串问题

题目大意

给定系数序列 f，定义 $w(S)$ 为 $f_{\frac{|S|}{d(S)}}$，其中 $d(S)$ 为 S 的最小整周期。

给定字符串 S，对每个 $1 \le i \le |S|$，求 $\sum_{j=1}^{i} w(S[j \cdots i])$。

$1 \le |S| \le 10^6$，5s。

QOJ15502 字符串问题

记 $T(S) = |S|/d(S)$ 为字符串 S 的周期数。容易发现，$\forall t \mid T(S)$，T 是一个 t 次方串。

我们希望枚举 $S[j \cdots i]$ 的周期数。先对 f 使用莫比乌斯反演求出 g 使得：$\sum_{d|i} g_d = f_i$。

这样枚举周期 t，把 g_t 加到答案里就行。g_1 是平凡的，考虑 $g_{\geq 2}$。进一步，我们转化为枚举串的长度，这样把 g 的一段前缀和加到答案里。

经典技巧：优秀的拆分。如果希望串长为 t，先按照 t 进行序列分块。容易发现，每个块内往前延续的最长只有 $O(1)$ 种可能，因此可以直接前缀和。

而如何计算往前延续的最长段？发现只需要计算 $O(n \log n)$ 次 LCP，使用二分 + 哈希可以做到 $O(n \log^2 n)$，使用 SA 可以做到 $O(n \log n)$。