Za 选串串 (A)

Problem A: [JSOI 2008] 火星人

维护一字符串 S，支持 M 次操作，每次操作为下列之一：

1. 询问两个后缀的 LCP 长度；
2. 修改一个字符；
3. 插入一个字符。

$\Sigma = \{\mathsf{a}, \dots, \mathsf{z}\}$，$M \le 1.5 \times 10^5$，$|S| \le 10^5$，询问不超过 10^4 个。

Source: https://loj.ac/p/2620

考虑维护哈希，则询问可以通过二分解决。

于是问题变为在可接受的时间复杂度内求子串哈希值。

这是容易的。解法包括但不限于：

- 平衡树；
- 离线后线段树；
- 对序列分块，块间维护链表；
- 对时间分块。

Problem B: [XR-3] 系统设计

给定一棵 n 个点的有根树，以及一个序列 $a_{1...m}$。

你需要支持 q 个操作，每次操作为下列之一：

- `1 x l r`：在 x 点放一枚棋子，枚举 $i = l \dots r$，每次将棋子移至其所在点的第 a_i 小的孩子处（若孩子数小于 a_i 则立即停止枚举），求最终棋子所在位置；
- `2 t k`：单点修改，$a_t \leftarrow k$。

$1 \le n, m, q \le 10^5$，$1 \le a_i \le n$，$1 \le x \le n$，$1 \le l \le r \le m$，$1 \le t \le m$，$1 \le k \le n$。

Source: https://www.luogu.com.cn/problem/P5537

记 $r_{1...n}$ 为每个点在兄弟姐妹间的排名。

对于每个点 x，记 s_x 为根到 x 的链上的 r 排成的序列。

则我们希望找到一个点 y，使得 $s_y = s_x + a_{l...r}$。

二分 + 哈希 + 哈希表即可。

Problem C: [HNOI 2004] L 语言

给定 n 个字符串构成的字典 $D = \{s_1, \ldots, s_n\}$ 以及 m 段文章 $t_{1...m}$。

你需要对每段文章，求出其最长的可被理解的前缀长度。

$1 \le n \le 20$，$1 \le m \le 50$，$1 \le |s_i| \le 20$，$1 \le |t_i| \le 2 \times 10^6$，$\Sigma = \{\mathtt{a} \ldots \mathtt{z}\}$。

可被理解的字符串，是指 D 中的一些可重的字符串拼接形成的任意字符串。如，$D = \{\mathtt{ab}, \mathtt{ba}\}$ 时，$\mathtt{abab}, \mathtt{abba}$ 可以被理解，而 $\mathtt{aba}, \mathtt{a}$ 是无法被理解的。

Source: https://www.luogu.com.cn/problem/P2292

设 $f_i \in \{0, 1\}$ 表示长为 i 的前缀能否被理解，则 $f_i = \vee f_j \ (t_{j-1...i} \in D)$。

由于 $|s_k| \le 20$，所以 f_i 只和 $f_{i-20...i-1}$ 有关，而后者可以状压进一个 `int` 中。

这启发我们可以利用位运算来优化转移。进而，我们只需要快速知道，$t_{1...i}$ 这一前缀的哪些后缀属于 D。这可以在 AC 自动机上预处理。

时间复杂度 $O(\sum |s_i| + \sum |t_i|)$。

Problem D: [EZEC-10] 序列

求有多少个不同的整数序列 $a_{1...n}$，满足下列条件：

- $0 \le a_i \le k \mid i = 1 \ldots n$;
- $a_{x_i} \oplus a_{y_i} = z_i \mid i = 1 \ldots m$，其中 $\oplus$ 表示按位异或。

答案对 10^9+7 取模。

$1\le n\le 5\times 10^5$，$0\le m\le 5\times 10^5$，$0\le z_i<2^{30}$，$1\le k<2^{30}$，$1\le x_i,y_i\le n$。

Source: https://www.luogu.com.cn/problem/P7717

将限制视作边，考虑每个连通块，若出现矛盾则必定无解。

对每个连通块，任意钦定一点为关键点 a_r，则对于任意与 r 连通的点 x，$a_r\oplus a_x$ 都会被若干限制唯一确定。

所以对每个连通块，问题变为：给定一集合 S，求有多少 $0\le v\le k$，满足 $\forall s\in S, 0\le v\oplus s\le k$。此时考虑 01-trie，则限制变为：在 trie 上某个节点，禁止向某一侧走。于是直接建个 trie 即可。

时间复杂度 $O(m+n\log v)$，其中 $v=\max\{x_i,y_i,z_i\}$。

Problem E: [CF 914 (Div. 1 + 2)] Substrings in a String

维护一个字符串 s，支持：

1. 单点修改；
2. 求 y 在 $s_{l\dots r}$ 中的出现次数。

$1\le |s|\le 10^5$，$1\le q\le 10^5$，$\sum |y|\le 10^5$，$\Sigma=\{\mathtt{a}\dots\mathtt{z}\}$。

Source: https://codeforces.com/problemset/problem/914/F

Solution A

考虑对 s 在空间上进行分块，设块长为 w。

满足 $|y|\ge w$ 的询问次数不超过 $\frac{|s|}{w}$，暴力 KMP 的复杂度为 $O(\frac{|s|^2}{w}+\sum|y|)$。

对于满足 $|y|<w$ 的询问，考虑 l,r 在 s 中的位置：

- 对于在散块中的出现次数，暴力 KMP 即可，复杂度为 $O(qw+\sum|y|)$。
- 对于跨过两块边界的出现次数，暴力 KMP 即可，复杂度为 $O(\frac{|s|}{w}\sum|y|)$。

- 问题仅剩下求出 y 在整块内部的出现次数。所有块的版本数之和是 $O(\frac{|s|}{w}+q)$ 的。对每个版本建出该版本上所有询问 y 的 AC 自动机进行匹配。处理一个版本的复杂度为 $O(w+|\Sigma|\sum_{(l,r,y)\in\text{Query}(\text{current version})}|y|)$。而一个 y 至多对 $O(\frac{|s|}{w})$ 个版本的复杂度产生贡献。所以总复杂度为 $O(|s|+qw+\frac{|s||\Sigma|}{w}\sum|y|)$。

总复杂度为 $O(\frac{|s|^2}{w}+qw+\frac{|s||\Sigma|}{w}\sum|y|)$。取 $w=\sqrt{\frac{|s|^2+|s||\Sigma|\sum|y|}{q}}$，可得最优时间复杂度 $O(\sqrt{q|s|(|s|+|\Sigma|\sum|y|)})$。

Solution B

考虑 `bitset`。记 O_c 为字符 c 的出现位置集合。

先初始化一个有且仅有 $l\dots r$ 为 1 的 `bitset` ans 表示答案。

然后枚举 $i=1\dots|y|$，令 $\text{ans}\leftarrow\text{ans}\cap(O_{y_i}-i+1)$，其中 $O_{y_i}-i+1$ 表示将 O_{y_i} 中所有元素减去 $i-1$ 后形成的 `bitset`。

时间复杂度 $O(\frac{|s|\sum|y|}{w})$。

Problem F: [NOI 2011] 阿狸的打字机

给定一棵字典树，同时给出 n 个终止节点，也即描述了 n 个字符串。

解决 m 次询问，每次询问形如：求第 x 个字符串在第 y 个字符串中的出现次数。

$1\le n\le 10^5$，$1\le m\le 10^5$，字典树大小 $\le 10^5$，$\Sigma=\{\texttt{a}\dots\texttt{z}\}$。

Source: https://loj.ac/p/2444

建出 AC 自动机，则 s 在 t 中出现了多少次，等价于在 s 对应节点的 fail 树子树中，有多少节点表示 t 的前缀。

考虑标记 t 的所有前缀对应的节点，查询就变成了子树求和。

标记节点可以在 trie 上按 DFS 序进行。

Problem G: [HAOI 2017] 字符串

认为两字符串 a, b 相似，当且仅当 $|a| = |b|$ 且 $\forall a_i \neq b_i, a_j \neq b_j$ 都有 $|i - j| < k$。

给定文本串 s 与 n 个模式串 $p_{1...n}$，分别求每个模式串在文本串中的相似子串数。

$|s|, \sum |p_i| \leq 2 \times 10^5$，字符集为 ASCII $33 \ldots 126$。

Source: https://loj.ac/p/2278

长度相等的两字符串 a, b 相似，等价于将 b 的某个长为 k 的区间替换为通配符后 $a = b$。可以利用类似点边容斥的思路，对每个 p_i，枚举长为 k 和 $k - 1$ 的区间替换为通配符，求出 p_i' 在 s 中的出现次数，即可加加减减求出答案了。

将 p_i 的某个长为 k 的区间替换为通配符后，会剩余一个前缀 pre 和一个后缀 suf。则我们要求的是有多少位置 j，满足 $s_{j...j+|\text{pre}|-1} = \text{pre}$ 且 $s_{j+|\text{pre}|+k...j+|\text{pre}|+k+|\text{suf}|-1} = \text{suf}$。

对所有模式串分别按前后缀建立 AC 自动机 $A_{\text{pre}}, A_{\text{suf}}$，然后记录文本串的每个位置在 AC 自动机上对应的节点。则问题变为，求有多少 i，满足 $s_{1...i+|\text{pre}|-1}$ 在 pre 的 fail 树子树内，且 $s_{i+|\text{pre}|+k...|s|}$ 在 suf 的 fail 树子树内。

将「在某子树内」的限制转化为「DFS 序在某区间内」，则问题变为二位数点。

Problem H: [NOI 2016] 优秀的拆分

对于一个字符串 S 的所有形如 $S = AABB$ 的拆分是优秀的，其中 A, B 非空。

给定一字符串 S，求其所有字串的优秀拆分数目之和。多测，T 组数据。

$\Sigma = \{\mathtt{a}, \ldots, \mathtt{z}\}$，$1 \leq T \leq 10$，$n \leq 3 \times 10^4$。

Source: https://uoj.ac/problem/219

对每个位置记录 f_i / g_i 表示以它为开头 / 结尾有多少 AA 串，则答案为 $\sum g_i f_{i+1}$。

问题变为求 f 和 g。枚举 $|A| = k$，将所有 k 的倍数的位置记为哨兵。则所有 AA 串中，两个 A 必定恰好包含一对相邻的哨兵。那么这对哨兵对应位置对应的前后缀必定满足 $|\text{LCP}| + |\text{LCS}| > k$。

于是枚举相邻哨兵对，不难通过 LCP, LCS 的长度计算 f 和 g。就做完了。

LCP, LCS 的长度可以二分 + 哈希，也可以后缀数组。

二分 + 哈希的总时间复杂度为 $O(\sum|S|\log^2|S|)$，后缀数组少个 $\log$。