3 QOJ1850 Magic Box

3.1 题目大意

给定一个长度为 n 的字符串 s。现需要从 s 中选出两个相等的子串 $s[l_1, r_1]$ 和 $s[l_2, r_2]$。

记一种选择方式的权值为满足 $l_1 \le i \le r_1, l_2 \le i \le r_2$ 的整数 i 的数量。对于所有 $0 \le k \le n$ 的整数 k，求有多少种权值为 k 的选择方式。

3.2 数据范围

$n \le 5 \times 10^5$，s 由小写英文字母组成。

3.3 解题过程

在下文中，记 $\text{lcp}(i, j)$ 表示 $s[i, n]$ 和 $s[j, n]$ 的最长公共前缀长度，记 $\text{lcs}(i, j)$ 表示 $s[1, i]$ 和 $s[1, j]$ 的最长公共后缀长度。

记 ans_k 表示权值为 k 的选择方式数目，则先求出 $\sum_{k=0}^{n} ans_k$ 表示选择 s 的任意一对相等子串的方案数。枚举一对 (l_1, l_2)，则合法的 (r_1, r_2) 个数即为 $\text{lcp}(l_1, l_2)$

建出 s 的后缀数组 [**SA**] 并求出 hei 数组表示排名相邻的后缀的 lcp，则排名为 $i, j (i < j)$ 的两个后缀的 lcp 即为 $\min_{i<x\le j} hei_x$。因此我们需要计算 hei 数组每一个区间的最小值之和，可以枚举 j，并用单调栈维护所有后缀的最小值之和，复杂度可以做到 $O(n)$。

求出 $\sum_{k=0}^{n} ans_k$ 之后，我们只需求出所有 $ans_k (1 \le k \le n)$，即可确定 ans_0 的值。

枚举 l_1, l_2（不妨设 $l_1 \le l_2$），记 $len = \text{lcp}(l_1, l_2)$，则合法的 (r_1, r_2) 需满足 $r_1 - l_1 + 1 \le len$，且 $r_1 - l_1 = r_2 - l_2$。另一方面，根据最长公共前缀的定义，这个条件是 (r_1, r_2) 合法的充分必要条件。

由于我们只关心 $k > 0$ 的答案，所以只需考察满足 $r_1 \ge l_2$ 的所有 (r_1, r_2) 即可。此时两个子串交的长度为 $r_1 - l_2 + 1$。换句话说，令 $d = l_2 - l_1$，则如果 $len \le d$，(l_1, l_2) 对答案没有贡献，否则对于所有 $d < k \le len$，会对 ans_{k-d} 产生 1 的贡献。

直接枚举 l_1, l_2，复杂度过高，不可接受。注意到贡献的下标中含有 d，所以我们转而枚举 $d = l_2 - l_1$。

记所有 $i = k \times d (k \in \mathbb{Z})$ 的位置 i 为关键位置，则对于任意 l_1，在区间 $(l_1, l_1 + d]$ 中都有恰好一个关键位置。我们枚举一个关键位置 i，然后计算所有 $i - d \le l_1 < i$ 的 l_1 对答案的贡献。

确定了 d, i 之后，记 $plen$ 表示 $\text{lcp}(i, i + d)$，$slen$ 表示 $\text{lcs}(i - 1, i + d - 1)$。由于一对 (l_1, l_2) 能对答案产生贡献的前提是 $\text{lcp}(l_1, l_2) \ge d$，且 $l_1 \ge i - d$，所以 $s[l_1, i] = s[l_1 + d, i + d]$。根据 lcs 的定义，即 $i - l_1 \le slen$。

另一方面，对于所有满足 $\max(i - d, i - slen) \le l_1$ 的 l_1，其一定满足 $s[l_1, i] = s[l_1 + d, i + d]$。此时在计算 $\text{lcp}(l_1, l_1 + d)$ 时，前 $i - l_1$ 个字符都是完全相同的，所以可以从 $i, i + d$ 这两个位置开始计算 lcp。即 $\text{lcp}(l_1, l_1 + d) = \text{lcp}(i, i + d) + i - l_1 = plen + i - l_1$。

现在我们枚举 d, i，可以通过预处理正串和反串的后缀数组 $O(1)$ 求出 $plen, slen$。此时只需对所有 $1 \le x \le \min(d, slen)$ 的 x，给所有 $0 < k \le plen + x - d$ 的 ans_k 都加上 1 即可。对 ans 做两次差分，可以转化为单点修改操作，即对于一对 (d, i)，计算的复杂度为 $O(1)$。

对于每个 d，关键点的数目为 $\frac{n}{d}$，所以总复杂度 $O(\sum_{i=1}^{n} \frac{n}{i}) = O(n \log n)$。预处理后缀数组的复杂度也是 $O(n \log n)$，故总复杂度 $O(n \log n)$。