

《研心》命题报告

福建省厦门双十中学 李静榕

题目大意

给定大小为 n 的字符串序列 S 和大小为 m 的字符串序列 T ，其中 S 的第 i 个字符串为 S_i ， T 的第 j 个字符串为 T_j 。

定义一个字符串的权值 $f(s)$ 为 s 中最长奇回文子串的半径长度。例如 `aba` 的半径长度为 2，`ababa` 的半径长度为 3。

定义两个字符串的加法 $s + t$ 为把两个字符串拼接起来得到的新字符串。

求 $\sum_{i=1}^n \sum_{j=1}^m f(S_i + T_j)$ 。

数据范围

令 $s = \max(\sum |S_i|, \sum |T_i|)$ 。

本题共有 4 个子任务，只有通过子任务中所有数据才能获得所有分数。

子任务编号	分数	特殊条件
1	20	$s \leq 5000$
2	30	$n = 1$
3	20	保证所有字符在 $\{a, b\}$ 中随机
4	30	无

对于 100% 的数据，满足 $1 \leq n, m, s \leq 4 \times 10^5$ ，保证输入的字符串只包含小写字母。

时间限制：5s

空间限制：2048MB

题解

部分分 1

暴力拼接所有串，用哈希或 manacher 算法检查最长奇回文串长度。

部分分 2

部分分 2 和部分分 3 的做法的前面部分在正解题解中。

因为 s 串已经固定，所以只需要统计回文中心在 s 串内的情况，其他可能的回文串长都可以提前算好，作为贡献挂在相应的 t 上。

通过数据结构手段（具体可见正解算法）可以算出回文中心在 s 串上时 t-trie 树上每个节点对应的 t 串（或其前缀）获得的最长回文子串的贡献。在 t-end（ t 串在 t-trie 上的结束位置）处求和即可。

同时存在一个根号做法，对于 $|t| > B$ 的串暴力查询最长回文，其他串添加到 trie 树中，就能不通过数据结构进行覆盖操作。

部分分 3

正解的数据结构部分会被大幅简化，标记可以暴力挂在节点上，最后执行计算答案的部分即可。

正解题解

首先考虑如何统计一个被缝合的串的奇回文长度。分成三段，一个是左边最长回文，右边最长回文，中间的回文。

中间的回文可以通过回文中心区分成左右两边，以左边为例：我们枚举左边串的回文后缀，将这个串去掉这个后缀后倒着对右侧匹配，匹配最长的是其中一个奇回文串，这里用二分哈希实现。匹配过程可以看作左边的串在右边的 m 个串形成的正 trie 进行匹配。

设左串个数为 n ，右串个数为 m ，考虑求出一个 $n \times m$ 的矩阵表示答案的矩阵，最终进行求和。

我们对于上述总共四种回文对矩阵进行覆盖（即 $chkmax$ ），例如左边最长回文是行覆盖，右边最长回文是列覆盖，回文中心在左侧是对一行进行一个 trie 树上的覆盖，回文中心在右侧是对一列进行一个 trie 树上的覆盖。

行和列的交互是非常困难的，首先考虑如何在只考虑行操作之后求出答案。

试图把一个可重的覆盖变成不重的覆盖，即多个“连续段”的值直接覆盖。单个左侧回文中心的操作具体是在 trie 树上对一条路径 S 进行如下操作：

设左回文后缀为 l ，那么对右侧 S 路径上深度为 d 的点 u 的子树进行关于 $2d + l$ 的 $chkmax$ 。这里 d 的系数直接钦定为 1（即原题求最长回文半径）。

提取出所有的 u 建立虚树，考虑所有在虚树上（包括长链上的）的 trie 树节点得到的权值。这个权值的意义是子树内所有 trie 树结束节点（例如固定左边那么 trie 树上就是 T 的 m 个结束节点）对应的最长回文对这个权值取 \max 。

考虑这么限制权值的值，使得每个点 u 的权值掌控（能真正更新 \max ）的部分是 u 的子树除去子树节点掌控的部分。进行两轮更新：

$$1. \text{chkmax}(val_{fa_u}, val_u + 1)$$

$$2. \text{chkmax}(val_u, val_{fa_u})$$

虚树上虚边对应的链的权值是形如 $1, 1, 1, 2, 3, 4$ 的形式，即一段相同权值接一段公差为 1 的等差数列。这样原 trie 树上每个点的权值都能被唯一覆盖，覆盖形式要不然是直接赋值（上面序列平的部分）或者是一个嵌套的结构，形式化地讲：设这条祖先-儿子方向的链为 C ，第 i 个点是 C_i ，那第 i 个点掌控的范围是 $subtree(C_i) - subtree(C_{i+1})$ ，权值是 $i + k$ ， k 是个关于这条链定值。

考虑把上述覆盖转到树剖上，其中第二种会拆成较多标记，忽略分讨过程之后，总的来说可以总结成三种：

1. 单点以权值 k 覆盖
2. 重链的一个区间的所有点和他们的轻儿子以权值 k 被覆盖
3. 重链的一个区间的所有点和他们的轻儿子以重链上的点的深度 $+k$ 为权值被覆盖

注意这种标记会有 $1/-1$ 的系数。当一个位置被权值为 $\{a_i\}$ （如 $\{1, -1, 2, -2, 3\}$ ）的标记覆盖，那么真正的标记权值是 $\sum a_i$ （上个例子是 3）。实现时保证标记集合可以通过消掉相反数的对最终得到至多一个正数标记。

然后进入纯 ds 部分。把树剖的重链拆成全局平衡二叉树，上述标记最终被拆成 $O(\log)$ 个子标记，每个标记都在一个全局平衡二叉树节点上，因为此时标记的类型相同（只需要考虑不同的 k ）。考虑 s 串产生的在 t-trie 上的标记和 t 串产生的在 s-trie 上的标记。将前者挂在 s-end（即 s-trie 的结束节点）上，它有对应的在 t-trie 全局平衡二叉树上的一个标记；将后者的标记部分挂在 s-trie 的全局平衡二叉树上，

它相应的有一个对应的 t-end。考虑特定的一对 s-t，我们想知道第一种标记和第二种标记的最大值。但是尽管上文提到这种标记是不重的覆盖，但实现中我们并不能做到真正的不重，是通过带系数的权值之间取 max 得到的。

具体的，这种标记可以看作 (val, k) 其中 k 表示系数。两个标记取 max 结果为 $(\max(val_1, val_2), k_1 k_2)$ 。可以证明这种标记具有交换律和结合律，更重要的，具有分配律。

对于这个特定的 s-t，两侧的标记实际上是标记集合，但是因为标记具有分配律，所以可以求出两个集合两两标记的最大值之后再求和。关注 s 的标记覆盖到 t-end 且 t 的标记覆盖到 s-end 的条件，枚举 s-trie 全局平衡二叉树的节点 u ，枚举所有 t 的标记覆盖在 u 上的标记，得到一种暴力做法：先枚举 u 覆盖下的节点 v （这个节点是原 s-trie 树节点），并枚举 v 作为 s-end 时所有对 t-trie 全局平衡二叉树的标记挂到这个二叉树节点 x 上，然后枚举标记覆盖 u 的对应的所有 t-end 节点 w ，枚举 w 的所有祖先和之前挂的标记取 max 后求和。因为查询一个 t-trie 全局平衡二叉树节点 x 上所有比某个标记大的标记的和需要二分，所以到这里最终复杂度为 $O(s \log^3 s)$ 。

实际上可以优化掉这个二分。首先考虑 s-end 对 t-trie 全局平衡二叉树的标记，放到同一个 x 上的标记需要排序，所以可以换成在 u 时先将所有 v 上的标记放到一起整体排序后按顺序添加标记，这个排序可以通过从 u 的儿子继承排序好的标记进行归并替换。对于 u 上标记对应的 w 向上查询的过程，我们先把查询挂在 w 上，并遍历 t-trie 全局平衡二叉树有查询的部分，从下至上向上归并查询，这样同时查询一个 x 的查询是排好序的，可以把标记和查询进行归并，同样求出 max 之和。总复杂度为 $O(s \log^2 s)$ 。

致谢

感谢中国计算机学会提供本次交流的平台。

感谢厦门双十中学施涵森同学提供 idea。

感谢陈旭磊、史家丞、仲煦北同学参与验题。

最后感谢任舍予学长、施涵森、陈睿泽、咸浩哲在 oi 生涯中对我的帮助、关心和支持。