

# L: 奶牛序列

## 题意

给出一个字符串，求公式

$$\sum_{1 \leq i < j \leq n} \text{len}(T_i) + \text{len}(T_j) - 2 * \text{lcp}(T_i, T_j)$$

$\text{lcp}$ 表示最长公共前缀。

字符串长度为 $5 * 10^5$

## 分析

学习了suffix array……

首先求出sa数组，然后就可以求出rank以及height数组，由lcp定理知道，

$\text{lcp}(i, j) = \min(\text{lcp}(k, k+1)) (i \leq k < j)$ , 于是就开始想奇奇怪怪的数据结构了。

### 1. solution1

维护一个线段树，枚举串 $i$ ，每个叶子节点表示前 $i-1$ 个串到第 $i$ 个串的区间最小值，也就是前面每个串和这第 $i$ 个串的lcp，然后进行一次query操作。想法完美，可惜复杂度太大，应该是常数太大（异常大），卡掉了。

### 2. solution2:

这个方法类似于LIS的nlogn做法。

- $\text{ans}[i]$ 表示到当前最后一个串的lcp中第 $i$ 大；
- $\text{con}[i]$ 表示前 $i$ 个串的lcp和；
- $\text{id}[i]$ 表示lcp第 $i$ 大的串的第一个id；
- $\text{sum}[i]$ 表示枚举到第 $i$ 个串的时候，仅仅由第 $i$ 个串贡献的 $\text{lcp}(j, i) (j < i)$

看更新好理解一些

枚举串 $i$ ，拿出 $\text{height}[i]$ 即其和前一个串的lcp，在 $\text{ans}$ 数组中二分查找

- 如果比 $\text{ans}$ 中任意一个都大，说明新增加的这个区间对之前的区间毫无影响，只是多了一个区间贡献 $\text{height}[i]$ ；
- 如果比 $\text{ans}$ 中任意一个都小，说明前面所有区间都被更新成了最小值；
- 如果更新在中间位置 $\text{pos}$ ，那么 $\text{pos}-1$ 对应的id之前的区间全都没有被修改，即该部分贡献仍然为 $\text{sum}[\text{id}[\text{pos}-1]]$ ，修改部分则全部变成了 $\text{height}[i]$ ，这个时候只需要加上区间个数 $\times \text{height}[i]$ 即可。而区间个数通过 $i - \text{id}[\text{pos}-1]$ 就可以得到。

## 思考

各种yy姿势……个人觉得自己的处理方法还不错。