

# 个人总结

## 陈铭煊 Max.D.

这次周四的I是个教训。提醒自己小心谨慎，以后口胡算法先仔细理解理解样例，然后写完代码七分钟之内不许马上提交，一定要好好造数据。

## 龙鹏宇 Hardict

稳定的罚时机器，每题先WA一发才能过

以后要多在本地造几组数据

需要在字符串方面多加练习

## 肖思炀 MountVoom

我，构造题废物，下周争取多做点构造题。

最近训练题数还行，就是做得有点慢以及罚时爆炸，需要注意罚时。

多练点题，提高想题速度。

# 本周推荐

## 陈铭煊 Max.D.

来源：

CF 666E [题目链接](#)

这周的比赛中出现了一道广义后缀自动机的题，居然还是没反应过来，因此再找一道有挑战性的EX\_SAM的题目来练练手。

标签：

广义后缀自动机，线段树合并

题意：

1. 给定一个字符串  $S$  和  $m$  个字符串  $T_1, T_2, \dots, T_m$
2. 有  $q$  个询问，每个询问给出四个参数  $l, r, p_l, p_r$
3. 求  $S$  的子串  $[p_l, p_r]$  在  $T_1, T_{l+1}, \dots, T_r$  中的哪个串出现的次数最多，
4. 如果出现次数最多的有多个串，取编号最小的，
5. 对于每组询问，输出编号和出现次数，
6.  $1 \leq |S| \leq 5 \times 10^5, 1 \leq m \leq 5 \times 10^4, 1 \leq \sum_{i=1}^m |T_i| \leq 5 \times 10^4, 1 \leq q \leq 5 \times 10^5$  时限  $6s$

### 题解：

假设已经有了一个  $S$  和  $T_1, \dots, T_m$  建立的广义后缀自动机。首先我们想找到  $[p_l, p_r]$  这个子串的对应节点，可以看做是  $[1, p_r]$  子串对应的节点，顺着自动机的树边（ $\text{Ink}$  指针组成的终止树）往根走找到，为了加速我们可以在这棵树上做个倍增。

假设我们找到的这个节点记载有记录出现的  $T_i$  的下标与次数的数据结构。我们所要做的，就是在这个数据结构里查找一个区间最值。

显然线段树是最合适不过了。在插入时我们只给新的  $\text{cur}$  指针上的线段树插入对应的  $T_i$  下标，最后我们从叶子往根走来做线段树合并，就可以得到每一个节点的记录线段树了。虽然不可能完整的开出线段树空间，但是考虑到我们插入和合并的次数不是那么多，用动态开点的方式就可以了。

这里要注意一个重要细节：对于字符串  $S$  我们只需要在自动机上爬点，为什么也需要将  $S$  插入自动机呢？事实上，不插入不行。假设我们跟着  $S$  在自动机上走  $[1, p_r]$  走到了一个点  $x$  那么  $x$  对应了一个字符串长度范围  $[\text{mLen}[\text{Ink}[x]]+1, \text{mLen}[x]]$  然而这个很可能  $[1, p_r]$  的  $[p_l, p_r]$  这一段才真正有匹配，且  $p_r - p_l < \text{mLen}[x]$  这样我们最后如果面对一个  $[p_l, p_r]$   $\text{mLen}[x] \geq p_r - p_l > p_r - p_l \geq \text{mLen}[\text{Ink}[x]]+1$  这样我们找到的点也还是  $x$  即使在  $x$  是没有对应匹配的字符串的。解决的方法就是插入  $S$  使得总能找到完全匹配的等长节点。

本题其实没有卡暴力找父亲，不倍增好像还快一丢丢。当然一般不要有这种侥幸心理。

AC代码：[982ms](#)，为了封装性写得有点略长。

### 评论：

还算可以的题，其实思维难度一般，主要还是写起来比较锻炼。列表里面好多人很久之前都写过这道陈年老题，有点惭愧。务必要多学习算法知识。

## 龙腾宇 Hardict

来源:

百度之星初赛[外部链接](#)

标签:

数论, 积性函数

题意:

$f(p^a) = p^a + 1, f(mn) = f(m)f(n), (m, n) = 1$   
 求  $\sum_{i=1}^N f(i), N \leq 10^{12}$

题解:

一开始化解出  $f()$  表达式后发现形式和  $\min 25$  筛差不多

然后学习了下最新的  $O(n^{\frac{2}{3}})$  的  $\min 25$  筛, 不过内存炸了

之后看题解才学习到了, 整体思路是参考杜教筛的推导过程

取积性函数  $g, h, s, t: f = g * h$

$h(1) = g(1) = 1, f(p^a) = \sum_{i=0}^a g(p^i) h(p^{a-i})$

$F(n) = \sum_{i=1}^n f(i) = \sum_{i=1}^n \sum_{j|i} h(j) g(\frac{i}{j})$   
 $= \sum_{j=1}^n \sum_{k=1}^{\frac{n}{j}} h(j) g(k)$   
 $= \sum_{j=1}^n h(j) G(\frac{n}{j}), G(n) = \sum_{i=1}^n g(i)$

注意到  $f(p) = g(p) + h(p)$ , 考虑令  $g(i) = \sum_{d|i} d = \sigma_1(i), h(p) = 0$

$f(p^2) = h(p^2) + (1 + p + p^2) = 1 + p^2, h(p^2) = -p$

$a > 2, f(p^a) = 1 + p^a = 1 + p^a + \sum_{i=3}^a h(p^i) g(p^{a-i}), h(p^a) = 0$

设  $r(n) = h(n^2), r(n) = \mu(n)n$

又考虑积性,  $F(n) = \sum_{i=1}^{\sqrt{n}} h(i^2) G(\frac{n}{i^2})$   
 $= \sum_{i=1}^{\sqrt{n}} \mu(i) i G(\frac{n}{i^2})$

评论: 对于积性函数, 除了直接套用各种模板性的现成公式

也可以学习模板推导过程, 进行自己的推导

## 肖思炀 MountVoom

构造题废物推荐一道构造题。

### A2. Prefix Flip (Hard Version) : 构造

题意:

给定两个 01 串  $s$  和  $t$  每次操作可以选择  $s$  的一个前缀进行翻转然后全部取反, 比如 0111 1110 0001。

需要在  $2n$  次操作以内把  $s$  串变为  $t$  串。

题解:

先说一下我的做法, 从后往前扫, 如果  $s[i] = t[i]$  则跳过, 如果  $s[i] \neq t[i]$  考虑  $s[1]$  如果  $s[1] \neq t[1]$  直接翻转, 否则先翻转  $s[1]$  再翻转前  $i$  位。这样做需要维护一下当前的  $s[1]$  和  $s[i]$  开始想维护整段甚至想上 splay (不至于不至于)。


然后在 zzh 鸽鸽的指点下, 这种题先考虑操作是否可逆, 即连着两次操作相当于没有操作。然后观察到我

们可以在n步以内把一个串变成全0或者全1，那么顺着操作一下s再逆着操作一下t就完事了。非常的简单好写。

comment□

注意考虑操作是否可逆，这样可以考虑“meet in middle”□把两边都转成同一个简单状态。

From:  
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:  
[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:alchemist:weekly\\_digest\\_7&rev=1595573013](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:alchemist:weekly_digest_7&rev=1595573013) 

Last update: **2020/07/24 14:43**