

# 简况

## 比赛链接

AC 5题, Rank 18th

# 总结与反思

## cmx

学了这么久广义后缀自动机还是不会C题（最后用其他方法勉强过了），思维还是需要训练。

## lpy

## xsy

# 题解

## C. Count New String

先说正解。

首先这题等价于求 $f(S, i, n)$ 这 $n$ 个串的不同子串的个数。

假设当前字符位置是 $i$ 最近的大于等于它的字符位置是 $j$ 那么我们可以在 $j$ 的基础上修改后缀自动机，插入的字符个数是 $j-i$ 所有的 $j-i$ 相加的个数不会超过 $10N$ 这是因为，如果把 $j$ 的条件弱化为等于 $S_i$ 的位置，那么这样的累和也不会超过 $10N$ 这是一个非常重要的性质。赛场上直观感受过，但是没有求出其上界。

这样，我们相当于是在一棵节点数最多为 $10N$ 的字典树上，走一遍广义后缀自动机。效率是 $O(N*10*10)$ 每插入一个节点，答案加上 $m_{len}[cur] - m_{len}[lnk[cur]]$ 即可。

比赛现场居然没往EX\_SAM的方向想，有点尴尬。

赛场上是xsy最后一个小时想出了一个比较极限的做法。求出每个 $f(S, i, n)$ 表示成一个十元组的形式（每个字符+出现次数），接着枚举每个 $f(S, i, n)$ 中间段，往一个以中间段为key的map中插入左右两边的字符数 $p$ 和 $q$ 表示该中间段两侧，左边可以加上 $[1..p]$ 个前一字符，右边可以加上 $[1..q]$ 个下一字符。注意，中间段最多有 $N*10*10$ 种

最后我们对每一个map的value(这是一个pair类型的vector)来求一次“矩形面积覆盖”，得到该中间段对应的不同字符串种类数，累和即可。

这么做最差是 $O(N*10*10*10*\log(N*10*10)+N*10*10*\log N)$  左边是构建map 右边是统计答案求和。

当然也可以不构建map 如果按照类似于基数排序或者字典树的方法插入中间段，那么可以把左边那个log给去掉。

其实效率还是很尴尬。不过最后赶时间交上去居然1A了，实际上，中间段的种类因为末尾的重复，会少很多。每个vector中矩形的个数也比满预期的少。

by cmx

## 补题

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

Permanent link:  
[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:alchemist:2020\\_nowcoder\\_multiuniversity\\_4&rev=1595309336](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:alchemist:2020_nowcoder_multiuniversity_4&rev=1595309336)

Last update: 2020/07/21 13:28

