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

想题写题都有点慢，需要多找找感觉。

需要自信一点，相信自己签到难度的数学题还是能做出来的qwq

题解

A. Ancient Distance

假设k是固定的，那么显然可以二分答案x然后每次选择一个最深的点把它的第x个祖先选为关键点，最后比较关键点和k的大小关系，一次检查可以利用线段树维护dfs序做到关键点个数$ \times \log$

而关键点的数量是$\leq \frac{n}{x + 1} + 1$的，而且随着答案的变大，需要的关键点的数量肯定是不增的。

考虑到枚举关键点数量再找答案效率较低，检查一次答案需要跑一次关键点个数$ \times \log$

而实际上我们一次检查其实可以求出每个答案最少需要的关键点的数量，于是可以枚举答案求最少关键点。

这样总复杂度就降为$O(n\log^2 n)$

by MountVoom

B. Basic Gcd Problem

签到题,

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$mlen[cur] - mlen[lnk[cur]]$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$O(N*10*10*10*log(N*10*10)+N*10*10*10*logN)$□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

效率甚至吊打了我去掉log的做法，自闭.jpg – xsy

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=1595407430

Last update: **2020/07/22 16:43**