

2020牛客暑期多校训练营（第四场）

[比赛链接](#)

B - Basic Gcd Problem

Solved by nikkukun.

水题不表。

C - Count New String

Upsolved by nikkukun.

题目描述

定义 $f(S, x, y)$ 是一个长度为 $y - x + 1$ 的串，且字符依次为 $S[x:x]$ 的最大值…… $S[x:y]$ 的最大值。求 $f(f(S, x_1, y_1), x_2, y_2)$ 有多少种本质不同的串，其中 $[x_2, y_2] \subseteq [x_1, y_1] \subseteq [1, n]$

解题思路 1

显然 $f(S, x_1, y_1)$ 必然是个串中字符递增的东西，然后递增东西里取 f 等于原串，故相当于问所有 $f(S, x_1, y_1)$ 有多少种本质不同的子串。同时枚举每个位置 x 发现实际上只需要求 $f(S, x, n)$ 而其他 $f(S, x, \text{last})$ 都是它的一个子串。因此，可以利用序列自动机求出 $f(S, i, n), i = 1, 2, \dots, n$ 这 n 个串本质不同的子串个数就是答案。

注意到这样的串可以用一个十元组唯一表示，因此可以枚举子串的开头结尾字符分别是十元组的哪两个位置，并按照这两个位置中间的子元组进行分类，计算出对一段中间固定的子元组，向左右添加字符可以构成什么串。这个问题相当于求二维平面上一些点向左下构成的矩形中，一共覆盖了多少个整点。

这里在比赛时写得非常混乱，是因为没有处理好边界重复的情况。如果设定好加在两端的字符都至少有一个，则这样子并不会出现重复（需要额外加上开头结尾是同一段的贡献）。另外，可以用 `vector` 做 key 方便代码编写。

总时间复杂度 $O(\Sigma^2 n \log n)$ 勉强能过。

解题思路 2

参考解题思路 1，如果能把 $f(S, i, n), i = 1, 2, \dots, n$ 全插到 SAM 里，那就能直接求答案，可惜字符串之和是 $O(n^2)$ 的。

考虑倒序建 trie 在 trie 上行走时，一旦有某个字母被新插入的字母更新了，则 trie 上会从原点向右多出来一条仅有一种字母的新链。对于每个位置而言，它只会被这样向右移动到新链上 $O(\Sigma)$ 次，故

trie 中最多 $O(\Sigma n)$ 个节点，于是可以跑广义 SAM 了。

总时间复杂度 $O(\Sigma^2 n)$

F - Finding the Order

Solved by nikukun.

题目描述

有平行线 $AB \parallel CD$ 或 $AB \parallel DC$ 给出 AC, AD, BC, BD 的长度，判断是哪种情况。

解题思路

可以先判断 C 和 D 在 AB 中线的哪一侧，然后分类讨论。标程的做法是 $AB \parallel CD$ 当且仅当 AD 或 BC 为四边中的最大值。

H - Harder Gcd Problem

Solved by qxforever.

题目描述

对 $1 \sim n$ 的数两两分组，使得组数尽可能多，且每组的两个数不互质，输出方案。

解题思路

显然大于等于 $\lceil n/2 \rceil$ 的质数和 1 无法分组，其他的数均可以分组。下面是一种方案。

对每个数，先塞到其最大质因数的桶中。对最大质因数 $p < \lceil n/2 \rceil$ 的桶，其大小至少为 2 。若为偶数，则桶内互相匹配；若为奇数，则将 $2p$ 扔到 2 的桶中，其他的数互相匹配。这样最多只有 1 个数不能匹配，是最优的。

老原题了

I - Investigating Legions

Solved by qxforever.

题目描述

有 n 个部队 m 个军团，每个部队属于一个军团。每个部队所属的军团是随机生成的。现在告诉你任意两个部队是否属于同一个军团，但是每条信息都有 $\frac{1}{S}$ 的概率是错误的。已知 $n \leq 300,20 \leq S \leq 100,1 \leq m \leq \lfloor n/30 \rfloor$

解题思路

注意到 $\frac{n}{m} \geq 30$ 说明每个军团都有约 30 个部队，且 S 比较大，不会显著的影响结果。

设当前最小的没有确定军团的部队为 x 选取所有 $a[x][i]=1$ 的部队。看这些部队之间的 a 是否相同，如果 $a[x][i]$ 是错误的信息，那么他和其他的部队关系总会是 0。之后再从 $a[x][i]=0$ 中找剩下的部分，看他和这些部队的关系是否总是 1 即可。单组数据时间复杂度为 $O(m \times n^2)$

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

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i_dont_know_png:multi2020-nowcoder-4&rev=1595289231

Last update: 2020/07/21 07:53

