Contest Info

date: 2020-08-03 12:00~17:00

2020牛客暑期多校训练营（第八场）

Solutions

D. Disgusting Relationship

题目大意：对于一个 n 和 $a_{1},\cdots,a_{n}$ 定义 $f(a_{1},\cdots,a_{n})$ 表示对每个 i 满足长为 i 的环恰有 a_{i} 个的 n 的不同排列数。求满足 $f(a_{1},\cdots,a_{n})$ 不能被 p（p 为质数）整除的 $\{a\}$ 的数量。

题解：首先求 f 这个比较简单，就是先枚举环的位置，乘上每个环内部的不同循环次数，最后再对相同长度去重。

$$ \begin{aligned} f(a_{1},\cdots,a_{n})&=\frac{n!}{\prod_{i=1}^{n}(i!)^{a_{i}}}\cdot\prod_{i=1}^{n}((i-1)!)^{a_{i}}\cdot\frac{1}{\prod_{i=1}^{n}a_{i}!}\\ &=\frac{n!}{\prod_{i=1}^{n}i^{a_{i}}\cdot\prod_{i=1}^{n}a_{i}!} \end{aligned} $$

从第一个等号可以看出，要使原式不被 p 整除，环长至多为 p 我们继续简化式子：

$$ \begin{aligned} &=\frac{n!}{\prod_{i=1}^{p}i^{a_{i}}\cdot\prod_{i=1}^{p}a_{i}!}\\ &=\frac{n!}{\prod_{i=1}^{p-1}i^{a_{i}}a_{i}!\cdot p^{a_{p}}a_{p}!} \end{aligned} $$

注意到对于 $1\sim p-1$ 它们的任意次方都不被 p 整除。因而原式中 p 的数量等价于下式：

$$ \frac{n!}{\prod_{i=1}^{p-1}a_{i}!\cdot(pa_{p})!} $$

从该式中可以直观地感受到 $2\sim p-1$ 的环数量不能太多，因为它们会导致分母中的 p 成比例减少，从而使得式子被 p 整除。

我们来证明 $t=a_{1}\bmod p+\sum_{i=2}^{p-1}ia_{i}<p$ 另外定义 $s=p\lfloor\frac{a_{1}}{p}\rfloor$ 考虑 $\frac{n!}{s!t!(pa_{p})!}$ 由于 $s+t+pa_{p}=n$ 该式为整数。我们证明在 $t\ge p$ 时，原式包含比该式更多的质因子 p 也即证明原式分母比该式包含严格更少的质因子 p

首先，若某个 $ia_{i}\ge p$ 考虑 $a_{i}!$ 和 $(ia_{i})!$ 容易证明前者到后者之间至少能出现一次质因子 p 命题已证毕。否则：

根据 kumor 定理 $a!b!$ 中 p 的数量不超过 $(a+b)!$ 中 p 的数量，少的个数为 $a+b$ 在 p 进制下发生进位的次数。因此

$$ \begin{aligned} &a_{1}!\cdot\prod_{i=2}^{p-1}a_{i}!\\ \le_{p}&s!\cdot(a_{1}\bmod p)!\cdot\prod_{i=2}^{p-1}ia_{i}!\\ <_{p}&s!t! \end{aligned} $$

最后一个地方不能取等号是因为，式中每一项均小于 p，而加起来的 t 却 $\ge p$，那么过程中必然发生了至少一次进位。

在该必要条件满足的前提下，由于 $p\mid pa_{p}$，t 只能等于 $n\bmod p$，容易证明，只要再满足 s 和 pa_{p} 做加法时不发生进位，就是充要条件了。

计数时，t 可以任意划分给 $1\sim p-1$，因此需要预处理划分数，$n-t$ 按位分配给 1 和 p 即可，比较简单。

时间复杂度 $\mathcal{O}(p\sqrt{p}+T\log n)$。

H. Hard String Problem

题目大意：给你 n 个串，但是每个串实际上是所给串的无限循环。求公共子串数量。

题解：首先注意到，每个串可以简化为最小循环，并转换为最小表示。相同的串可以删除。如果此时只剩一个串了，那么自然有无限个串。否则，考虑最短的串 s 和 t，如果它们有一个长度 $\ge|s|+|t|$ 的公共子串，根据弱周期引理，$\gcd(|s|,|t|)$ 也是它们的周期，那么假如 $|s|\neq|t|$，与最小循环矛盾；否则与 $s\neq t$ 矛盾。因此公共子串最长为 $|s|+|t|-1$，这时将每个串循环几遍，使得包括所有这么长的子串即可。容易发现，除 s,t 外的串长度至多变为原来的 3 倍，因而仍能保证总串长，变为了经典 SAM 题。

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

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:intrepidsword:2020-nowcoder-multi-8

Last update: **2020/08/05 11:45**