OI 数学基础

Owaski

雅礼中学

Preface

今天来讲一下 OI 中出现过的一些数学方面的东西，听过的就当复习一遍咯，不要干扰别人听课，没听过的就当新课听咯，都是一些基础知识。

(你如果要睡觉我也没办法 QAQ)

Contents

1. 数论基础
2. 数论函数
3. 组合相关
4. 有标号图的计数

带余除法

对于两个整数 a, b，存在两个唯一的整数 q, r 满足：

$$b = aq + r, (0 \leq r < |a|)$$

取整

$\lfloor x \rfloor$ 和 $\lceil x \rceil$ 分别是向下和向上取整。
显然带余除法中的 q 就是 $\left\lfloor \frac{a}{b} \right\rfloor$。

整除

整数 a, b，且 $a \neq 0$，如果存在整数 $b = k \cdot a$，那么 $a|b$，a 整除 b。b 称为 a 的倍数，a 称为 b 的约数 (因数)。

一道经典题

$$\sum_{i=1}^{n} \left\lfloor \frac{n}{i} \right\rfloor, n \le 10^9$$

最大公约数，最小公倍数

最大公约数 gcd，最小公倍数 lcm。

辗转相除法：$\gcd(a,b)=\gcd(b,a \bmod b)$。

gcd 和 lcm 关系：$\operatorname{lcm}(a,b)=\frac{ab}{\gcd(a,b)}$，这个关系无法拓展到多个数。

扩展欧几里得

给定 a, b, c，求 $ax + by = c$，x, y 的解。

noip 内容，只有当 $\gcd(a, b) | c$ 时有解。

类欧几里得算法

给定 n, a, b, c，求：

$$\sum_{i=0}^{n} \left\lfloor \frac{a \cdot i + b}{c} \right\rfloor$$

$n \le 10^{18}$

类欧几里得算法

很显然我们可以通过取模把 $a < c$。

$$\begin{aligned}
\sum_{i=0}^{n}\left\lfloor\frac{a\cdot i+b}{c}\right\rfloor &= \sum_{i=0}^{n}\sum_{j<\left\lfloor\frac{a\cdot i+b}{c}\right\rfloor} 1 \\
&= \sum_{j=0}^{\left\lfloor\frac{a\cdot n+b}{c}\right\rfloor-1}\sum_{i=0}^{n}[j<\left\lfloor\frac{a\cdot i+b}{c}\right\rfloor] \\
&= \sum_{j=0}^{\left\lfloor\frac{a\cdot n+b}{c}\right\rfloor-1}\sum_{i=0}^{n}[i>\left\lfloor\frac{c\cdot j+c-b-1}{a}\right\rfloor] \\
&= \sum_{j=0}^{\left\lfloor\frac{a\cdot n+b}{c}\right\rfloor-1} n-\left\lfloor\frac{c\cdot j+c-b-1}{a}\right\rfloor
\end{aligned}$$

类欧几里得算法

分母变成了 a，这个过程和欧几里得算法的递归是一个性质的，因此复杂度也是 $O(\log n)$

质数

质数就是只有 1 和本身是自己的因数的数。

n 以内质数的出现频率是 $O(\frac{n}{\log n})$ 的。

判断质数

枚举 $1 \sim \sqrt{n}$ 内的数。

或者 $O(k \log n)$ 的 miller-rabin 算法，k 表示选取的检测的数的个数。

大数质因数分解

pollard-rho && miller-rabin

复杂度 $O(n^{1/4}\log n)$

求出 $1 \sim n$ 的所有质数

$O(n\sqrt{n})$ 的暴力判断方法。

$O(n\log n)$ 的枚举倍数方法。

以及 $O(n)$ 的欧拉筛法。

又一个经典问题

如何快速求 $1 \sim n$ 的质数个数呢？

$O(n)$ 的筛法自然不必说，有没有低于 $O(n)$ 的做法呢？

又一个经典问题

如何快速求 $1 \sim n$ 的质数个数呢？

$O(n)$ 的筛法自然不必说，有没有低于 $O(n)$ 的做法呢？

考虑 $\pi(n)$ 表示 $1 \sim n$ 的质数个数。

设 $f(n, m)$ 表示 $[1, n]$ 内不包含 $p_1 \cdots p_m$ 的数的个数，那么有：

$$f(n, m) = f(n, m-1) - f(\lfloor n/p_m \rfloor, m-1)$$

又一个经典问题

考虑设 $k = n^{1/3}$，设 $P_r(n, m)$ 表示 $1 \sim n$ 内最小质因子 $> p_m$ 且有 r 个质因子的数的个数，那么有:

$$\pi(n) = \pi(k) + f(n, \pi(k)) - 1 - P_2(n, \pi(k))$$

$$P_2(n, m) = \sum_{p_m < k \leq n^{1/2}, k\text{是质数}} \pi(\lfloor n/k \rfloor) - \pi(k) + 1$$

计算 f 的复杂度是 $O(\pi(k)\sqrt{n})$ 的，计算 $P_2(n, \pi(k))$ 的复杂度是 $O(\lfloor n/k \rfloor)$ 的。

总复杂度就是 $O(\frac{n^{5/6}}{\log n})$ 的。

还可以更快。

又一个经典问题

令 $k = n^{1/4}$，那么我们还需要计算的就是 $P_3(n, \pi(k))$。

$$P_3(n, m) = \sum_{p_m < k \le n^{1/3}, k\text{是质数}} P_2(\lfloor n/k \rfloor, \pi(k-1))$$

计算一次 $P_2(n, m)$ 的复杂度就是 $O(n^{1/2})$ 的，所以：

$$\sum_{k \le n^{1/3}, k\text{是质数}} \sqrt{\lfloor n/k \rfloor} \le n^{2/3}$$

因此总复杂度是 $O(n^{3/4})$ 的。

其实还可以更快，the Meissel, Lehmer, Lagarias, Miller, Odlyzko method 有兴趣的可以课后了解一下。

取模

带余除法中的 $b \bmod a = r$，也可以写成 $r \equiv b \pmod{a}$。

快速幂，快速乘

$a^b \bmod m$

$a, b, m \leq 10^{18}$。

noip 内容。

快速幂，快速乘

$a^b \bmod m$

$a, b, m \leq 10^{18}$。

noip 内容。

还有一种卡常数的 long double 乘法，在这里就不讲了。(正常出题人都不会用这个卡常数 = =)

还是一道经典题

$$\sum_{i=1}^{n} n \bmod i,\ n \le 10^9$$

模意义下乘法逆元

对于整数 a, m，如果存在 x 满足 $ax \equiv 1 \pmod m$，那么 x 是 a 在模 m 下的乘法逆元。

乘法逆元可以用扩展欧几里得求，解方程 $ax + bm = 1$ 即可。

如果 $\gcd(a, m) = 1$ 也可以用欧拉定理，$a^{\varphi(m)} \equiv 1 \pmod m$

依旧是一道经典题

已知 a, b, m，$ax \equiv b \pmod{m}$ 求 x

依旧是一道经典题

已知 a, b, m， $ax \equiv b \pmod m$ 求 x

显然 $\gcd(a, m) \nmid b$ 的时候无解，所以这种时候就不要考虑除法了。

除掉 $\gcd(a, m)$ 后，现在满足 $\gcd(a, m) = 1$，而且问题可以简化成 $b = 1$ 的情况，所以问题变成了求 $ax \equiv 1 \pmod m$，求 a 的逆元，直接上扩展欧几里得。

中国剩余定理 (CRT)

对于 x，我们知道 $x \bmod m_1, x \bmod m_2, \cdots, x \bmod m_r$，且满足 $\forall_{i,j} \gcd(m_i, m_j) = 1$，那么我们可以算出 $x \bmod m, m = \prod_{i=1}^{r} m_i$。

中国剩余定理 (CRT)

对于 x，我们知道 $x \bmod m_1, x \bmod m_2, \cdots, x \bmod m_r$，且满足 $\forall_{i,j} \gcd(m_i, m_j) = 1$，那么我们可以算出 $x \bmod m, m = \prod_{i=1}^{r} m_i$。

构造答案的过程很简单，考虑对于某个 i，设:

$$M(i) \equiv \frac{\prod_{j=1}^{r} m_j}{m_i} \pmod m, N(i) \equiv M(i)^{-1} \pmod {m_i}$$

那么我们要的结果就是:

$$\sum_{i=1}^{r} M(i) \cdot N(i) \cdot x_i$$

组合数取模

$$\binom{n}{m} \bmod p$$

- version 1 $n, m \leq 10^5$
- version 2 $p \leq 10^5$ 且 p 是质数,
- version 3 $p = \prod_{i=1}^{r} p_i^{k_i}$ 且 $\sum_{i=1}^{r} p_i^{k_i} \leq 10^7$

version 1

$O(n)$ 预处理组合数，$O(1)$ 回答询问。

version 2

lucas 定理

$$\binom{n}{m} \equiv \binom{\lfloor n/p \rfloor}{\lfloor m/p \rfloor} \binom{n \bmod p}{m \bmod p} \pmod p$$

预处理 $O(p)$，询问 $O(\log n)$。

version 3

version 3

首先考虑将 p 质因数分解，之后答案用 CRT 合并即可，现在考虑模数为 p^k,p 为质数。首先我们知道：

$$\binom{n}{m} = \frac{n!}{m!\,(n-m)!}$$

我们可以把 $n!$ 表示成 $a \cdot p^b$ 的形式，其中 $\gcd(a,p) = 1$，这样我们就能直接做阶乘的除法了，比如说 $n! = a \cdot p^b, m! = c \cdot p^d$，那么有：

$$\frac{n!}{m!} \equiv a \cdot c^{-1} \cdot p^{b-d} \pmod{p^k}$$

因为 $\gcd(c,p) = 1$，因此我们可以直接求逆元。现在考虑怎么把 $n!$ 表示成 $a \cdot p^d$ 的形式。

version 3

考虑把 $1 \cdots p^k - 1$ 范围内不为 p 的倍数的数给拿出来，因为这些数与 p 的 gcd 肯定为 1，因此都可以直接算到 a 的贡献里去，剩下的数为 $p, 2p, 3p, \ldots, \left\lfloor \frac{n}{p} \right\rfloor \cdot p$，都含有因子 p，因此我们可以提取出一个 p，给 b 贡献 $\left\lfloor \frac{n}{p} \right\rfloor$，然后递归进去算 $\left\lfloor \frac{n}{p} \right\rfloor! \pmod{p^k}$ 的值。

算上 CRT 的复杂度，预处理阶乘我们就得出了 $O(\sum p_i^{k_i})$ 的复杂度的算法。

欧拉定理

对于正整数 n，令 $\varphi(n)$ 表示比 n 小的与 n 互质的数的个数，有：

$$\forall \gcd(a,n)=1, a^{\varphi(n)} \equiv 1 \pmod n$$

其中 $\varphi(n)$ 称为欧拉函数。

欧拉定理的证明

欧拉定理的证明

考虑所有小于 n 且与 n 互质的数 $x_1, x_2, \cdots, x_{\varphi(n)}$，对每个 x_i 乘一个 a，因为 $\gcd(a, n) = 1$，因此 $x_i \cdot a \not\equiv x_j \cdot a \pmod n, i \neq j$。

又因为 $\gcd(x_i, n) = \gcd(a, n) = 1$，所以 $\gcd(x_i \cdot a, n) = 1$，所以乘 a 后的集合是不变的，因此：

$$\prod_{i=1}^{\varphi(n)} x_i \cdot a \equiv a^{\varphi(n)} \prod_{i=1}^{\varphi(n)} x_i \equiv \prod_{i=1}^{\varphi(n)} x_i \pmod n$$

得出 $a^{\varphi(n)} \equiv 1 \pmod n$ 的结论。

欧拉定理的扩展

欧拉定理要求 $\gcd(a, n) = 1$，那 $\gcd(a, n) \neq 1$ 的时候是否成立呢？

欧拉定理的扩展

欧拉定理要求 $\gcd(a, n) = 1$，那 $\gcd(a, n) \neq 1$ 的时候是否成立呢？
有个公式是这样的：

$$a^m \equiv a^{\min\{m, m \bmod \varphi(n) + \varphi(n)\}} \pmod{n}$$

可以形象地理解成一个 ρ 形循环。

一道玲珑杯题

给定一个长度为 n 的序列 a 和一个整数 p，问所有 a 的排列中，最小的下面式子的值是多少。

$$a_1^{a_2^{\cdot^{\cdot^{\cdot^{a_n}}}}} \mod p$$

$1 \leq n \leq 8, 1 \leq p \leq 10^9, 1 \leq a_i \leq 10^9$

一道玲珑杯题

给定一个长度为 n 的序列 a 和一个整数 p，问所有 a 的排列中，最小的下面式子的值是多少。

$$a_1^{a_2^{\cdot^{\cdot^{\cdot^{a_n}}}}} \mod p$$

$1 \le n \le 8, 1 \le p \le 10^9, 1 \le a_i \le 10^9$

$n!$ 枚举排列，然后考虑利用扩展的欧拉定理，计算 $\varphi(n), \varphi(\varphi(n)), \cdots$ 即可。

一道 PoPoQQQ 题

求下列式子的值

$$2^{2^{2^{\cdot^{\cdot^{\cdot}}}}} \mod p$$

1000 组数据，$p \leq 10^7$

一道 PoPoQQQ 题

求下列式子的值

$$2^{2^{2^{\cdot^{\cdot^{\cdot}}}}} \bmod p$$

1000 组数据，$p \le 10^7$

令 $p = q \cdot 2^k$，其中 q 为奇数，答案可以表示成 $2^k(2^{2^{2^{\cdot^{\cdot^{\cdot}}}}} - k \bmod q)$，这时可以用欧拉定理，对 $2^{2^{2^{\cdot^{\cdot^{\cdot}}}}} \bmod \varphi(q)$，然后一直递归下去，直到 $\varphi(q) = 1$ 返回 0 即可，最长链也只有 24。可以在递归时计算 $\varphi(q)$，也可以预处理 (实践证明递归时计算还快些)。

阶

对于两个互质的整数 a, m，定义 a 对模 m 的阶为最小的正整数 d 满足：

$$a^d \equiv 1 \pmod{m}$$

记作 $\delta_m(a)$。

阶

对于两个互质的整数 a, m，定义 a 对模 m 的阶为最小的正整数 d 满足：

$$a^d \equiv 1 \pmod m$$

记作 $\delta_m(a)$。

显然 $\delta_m(a) \mid \varphi(m)$，是个 O 形循环。

阶

对于两个互质的整数 a, m，定义 a 对模 m 的阶为最小的正整数 d 满足：

$$a^d \equiv 1 \pmod m$$

记作 $\delta_m(a)$。

显然 $\delta_m(a) \mid \varphi(m)$，是个 O 形循环。

定理：$\delta_m(a^k) = \frac{\delta_m(a)}{\gcd(\delta_m(a),k)}$。

推论：$\gcd(\delta_m(a), k) = 1$ 时，$\delta_m(a^k) = \delta_m(a)$。

求阶

求阶

因为 $\delta_m(a) \mid \varphi(m)$，暴力枚举 $\varphi(m)$ 的约数即可。

原根

满足 $\delta_m(a) = \varphi(m)$ 的 a 称为 m 的原根。

原根一旦有就有 $\varphi(\varphi(m))$ 个。

质数一定有原根。

求原根

求原根

暴力枚举咯，枚举 a，枚举 $\varphi(m)$ 的质因子 p，判断是否存在 $a^{\varphi(m)/p} \equiv 1 \pmod m$，存在 a 就不是原根。

一道 SDOI 题

小 C 有一个集合 S，里面的元素都是小于 M 的非负整数。他用程序编写了一个数列生成器，可以生成一个长度为 N 的数列，数列中的每个数都属于集合 S。小 C 用这个生成器生成了许多这样的数列。但是小 C 有一个问题需要你的帮助：给定整数 x，求所有可以生成出的，且满足数列中所有数的乘积 $\bmod M$ 的值等于 x 的不同的数列的有多少个。小 C 认为，两个数列 $\{A_i\}$ 和 $\{B_i\}$ 不同，当且仅当至少存在一个整数 i，满足 $A_i \neq B_i$。另外，小 C 认为这个问题的答案可能很大，因此他只需要你帮助他求出答案 $\bmod 1004535809$ 的值就可以了。

$M \leq 8000$，M 为质数，$N \leq 10^9$，$1 \leq x \leq M-1$

一道 SDOI 题

因为 M 是质数，我们可以求出它的原根，设为 g，我们可以把 S 每个元素表示成 g^k，那么问题转变成了加法，设加法集合为 $a_1, a_2, \cdots, a_r$，那么母函数就是：

$$F = \sum_{i=1}^{r} x^{a_i}$$

求出 F^N，令 $x \equiv g^k \pmod M$，F^N 的所有 $\bmod M - 1 = k$ 的位置的和就是答案，可以在多项式快速幂的时候做一个循环卷积，注意并不需要你真正实现循环卷积，可以 NTT 之后把对应的位置加起来。

离散对数

对于 a, b, m，其中 a, b 与 m 互质，定义 $\log_{m,b}(a)$ 为最小的 d 满足：

$$a^d \equiv b \pmod m$$

显然所有 d 模 $\delta_m(a)$ 同余。

求离散对数

baby-step giant-step，俗称大步小步法。

设一个参数 l，将 $a^0, a^1, a^2, \cdots, a^{l-1}$ 存入 hash 表，然后枚举 $k \le \frac{\varphi(m)-1}{l}$，每次查询 hash 表中是否有 j 满足 $a^{kl+j} \equiv b \pmod{m}$，即 $b \cdot a^{-kl} \bmod m$，复杂度 $O(\sqrt{\varphi(m)})$，当然也可以根据具体情况调整 l 的大小。

如果 a, b 不与 m 互质呢？

EXTBSGS

原理很简单，我们需要把 a, m 变成互质的，每次取 $d = \gcd(a, m)$，如果 $d \nmid b$，显然无解，否则把方程全部除掉一个 d，递归下去直到 $d = 1$。

一道玲珑杯题

现在有一个 $n \times n \times n$ 的立方体，每个格子表示成 (x, y, z) 的三元组。每次操作，将 (x, y, z) 位置上的值移到 $((x + y + z) \bmod n, (x + 2y + 2x) \bmod n, (2x + 3y + 4z) \bmod n)$ 处，可以证明这是个双射。

现在要你求出最少经过多少步立方体会变回原来的样子。

$n \leq 10^5$

一道玲珑杯题

考虑一个转移矩阵 A，使得：

$$(x, y, z) \times A = ((x+y+z) \bmod n, (x+2y+2x) \bmod n, (2x+3y+4z) \bmod n)$$

那么问题变成了求出一个 x 使得：

$$A^x = I \pmod n$$

BSGS 套上矩阵求逆即可。

常见的数论函数

- $e(n) = [n = 1]$
- $id(n) = n$
- $1(n) = 1$
- $\sigma_k(n) = \sum_{d|n} d^k$
- $\varphi(n) = \sum_{i=1}^{n} [\gcd(i, n) = 1]$
- $\mu(n) = \begin{cases} 0, \exists d > 1, d^2 | n \\ (-1)^k, n = \prod_{i=1}^{k} p_i \end{cases}$

积性函数

若 $\gcd(a,b)=1$，$f(ab)=f(a)f(b)$，那么 f 是积性函数。

如果对于任意 a,b 满足 $f(ab)=f(a)f(b)$，那么 f 是完全积性函数。

积性函数

若 $\gcd(a,b)=1$，$f(ab)=f(a)f(b)$，那么 f 是积性函数。

如果对于任意 a,b 满足 $f(ab)=f(a)f(b)$，那么 f 是完全积性函数。

一个有趣的事实是，$g(n)=\sum_{d|n}f(d)$，如果 $g(n)$ 是积性函数，那么 $f(n)$ 也是积性函数，证明可以使用归纳法。

φ

φ 之前我们已经接触过了，考虑 φ 的几种性质：

$$\varphi(p) = p - 1$$

$$\varphi(p^k) = p^k - p^{k-1} = p^{k-1}(p-1) = p^{k-1}\varphi(p)$$

$$\varphi(a \cdot b) = \varphi(a)\varphi(b), \gcd(a, b) = 1$$

$$\varphi(n) = \prod_{i=1}^{r} \varphi(p_i^{r_i}) = n\prod_{i=1}^{r}(1 - \frac{1}{p_i})$$

那么我们求 φ 就可以用线性筛或者分解质因数了。

φ

另一个特殊的性质：

$$n = \sum_{d|n} \varphi(d)$$

φ

另一个特殊的性质：

$$n = \sum_{d|n} \varphi(d)$$

一个有意思的证明是考虑 $\frac{1}{n}, \frac{2}{n}, \cdots, \frac{n}{n}$，将它们化成最简分数之后，分母肯定都是 n 的约数，且分子分母是互质的，等价于所有 n 的约数的 φ 的和。

一道经典题

$$\sum_{i=1}^{n} \gcd(i, n)$$

$$n \leq 10^9$$

一道经典题

$$\sum_{i=1}^{n} \gcd(i, n)$$

$$n \leq 10^9$$

$$\sum_{i=1}^{n} \gcd(i, n) = \sum_{d|n} d \cdot \varphi(\frac{n}{d})$$

加强版

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\gcd(i,j)$$

$$n, m \le 10^7$$

加强版

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\gcd(i,j)$$

$$n,m \le 10^7$$

$$\begin{aligned}\sum_{i\le n}\sum_{j\le m}\gcd(i,j) &= \sum_{i\le n}\sum_{j\le m}\sum_{d\mid i,j}\varphi(d)\\ &= \sum_{d\le n,m}\varphi(d)\left\lfloor\frac{n}{d}\right\rfloor\left\lfloor\frac{m}{d}\right\rfloor\end{aligned}$$

再谈 $\varphi(ab)$

如果 a, b 不互质啥情况呢？

再谈 $\varphi(ab)$

如果 a, b 不互质啥情况呢？

- $$\varphi(ab) = \varphi(a)\varphi(b)\frac{d}{\varphi(d)}, d = \gcd(a, b)$$

- 推论：
$$\varphi(a)\varphi(b) = \varphi(\operatorname{lcm}(a, b))\varphi(\gcd(a, b))$$

μ

莫比乌斯函数 μ 的定义式：

$$\sum_{d|n} \mu(d) = [n = 1]$$

μ

莫比乌斯函数 μ 的定义式：

$$\sum_{d|n} \mu(d) = [n = 1]$$

因为 $[n = 1]$ 是个积性函数，所以 μ 也是积性函数。

$$\mu(p^k) = \begin{cases} 1, k = 0 \\ -1, k = 1 \\ 0, k > 1 \end{cases}$$

莫比乌斯反演

$$f(n)=\sum_{d|n}g(d)\Leftrightarrow g(n)=\sum_{d|n}\mu(d)f(\frac{n}{d})$$

莫比乌斯反演的证明

$$\begin{aligned}\sum_{d|n} g(d) &= \sum_{d|n}\sum_{p|d}\mu(p)f(\frac{d}{p}) \\ &= \sum_{d|n} f(d)\sum_{p|\frac{n}{d}}\mu(p) \\ &= f(n)\end{aligned}$$

反过来差不多，这里就不写了。

这玩意儿还有扩展形式

$$g(n)=\sum_{d\geq 1} f(\frac{n}{d}) \Leftrightarrow f(n)=\sum_{d\geq 1} \mu(d) g(\frac{n}{d})$$

$$g(n)=\sum_{n|d} f(d) \Leftrightarrow f(n)=\sum_{n|d} g(d) \mu(\frac{d}{n})$$

φ && μ

$$n = \sum_{d|n} \varphi(d), \varphi(n) = \sum_{d|n} \mu(d) \frac{n}{d}$$

φ && μ

$$n=\sum_{d|n}\varphi(d),\varphi(n)=\sum_{d|n}\mu(d)\frac{n}{d}$$

$$\sum_{i\le n}\varphi(i)=\sum_{i\le n}\sum_{d|i}\mu(d)\frac{i}{d}=\sum_{d\le n}\mu(d)\sum_{i\le\left\lfloor\frac{n}{d}\right\rfloor}i=\frac{1}{2}\left(1+\sum_{i\le n}\mu(i)\left\lfloor\frac{n}{i}\right\rfloor^2\right)$$

$$\sum_{i\le n}\frac{\varphi(i)}{i}=\sum_{i\le n}\sum_{d|i}\frac{\mu(d)}{d}=\sum_{d\le n}\frac{\mu(d)}{d}\left\lfloor\frac{n}{d}\right\rfloor$$

$1 \sim n$ 中与 n 互质的数的和

$1 \sim n$ 中与 n 互质的数的和

$$
\begin{aligned}
S(n) &= \sum_{i=1}^{n} i \cdot [\gcd(i, n) = 1] \\
&= \sum_{d \mid n} \mu(d) \sum_{i=1}^{\frac{n}{d}} id \\
&= \sum_{d \mid n} \mu(d) \frac{n(\frac{n}{d} + 1)}{2} \\
&= \frac{n}{2}(\varphi(n) + [n = 1])
\end{aligned}
$$

φ, μ 前缀和

φ, μ 前缀和

$$\phi(n) = \sum_{i=1}^{n} \varphi(i) = \sum_{i=1}^{n} (i - \sum_{d|i, d \neq i} \varphi(d)) = \frac{n(n+1)}{2} - \sum_{i=2}^{n} \phi(\left\lfloor \frac{n}{i} \right\rfloor)$$

$$M(n) = \sum_{i=1}^{n} \mu(i) = \sum_{i=1}^{n} ([i = 1] - \sum_{d|i, d \neq i} \mu(d)) = 1 - \sum_{i=2}^{n} M(\left\lfloor \frac{n}{i} \right\rfloor)$$

预处理前 $n^{2/3}$ 项，更大的范围递归，$O(n^{2/3})$。

前面东西的抽象形式

对于一个函数 f，如果存在函数 g，满足 $g(n)=\sum_{d|n} f(n)$，那么 f 的前缀和 F 可以写成：

$$F(n)=\sum_{i=1}^{n} f(i)=\sum_{i=1}^{n}(g(i)-\sum_{d|i,d\neq i} f(d))=\sum_{i=1}^{n} g(i)-\sum_{i=2}^{n} F(\left\lfloor\frac{n}{i}\right\rfloor)$$

可以发现前提是 g 的前缀和得要好算才行，这就很尴尬了。当然不一定所有题都是这种形式，有时候需要变形。

这玩意俗称杜教筛。

来练一道题

$$S(n)=\sum_{i=1}^{n}\sum_{j=1}^{n}\operatorname{lcm}(i,j)$$

$$n\le 10^{10}$$

推推推

$$\begin{aligned}
S(n) &= \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{ij}{\gcd(i,j)} \\
&= \sum_{d=1}^{n} d \sum_{i=1}^{\lfloor \frac{n}{d} \rfloor}\sum_{j=1}^{\lfloor \frac{n}{d} \rfloor} ij[\gcd(i,j)=1] \\
&= \sum_{d=1}^{n} d \sum_{g=1}^{\lfloor \frac{n}{d} \rfloor} \mu(g)\cdot g^2 \cdot f(\left\lfloor \frac{n}{d\cdot g} \right\rfloor) \\
&= \sum_{g=1}^{n} \mu(g)\cdot g^2 \sum_{d=1}^{\lfloor \frac{n}{g} \rfloor} d\cdot f(\left\lfloor \frac{n}{d\cdot g} \right\rfloor)
\end{aligned}$$

其中 $f(n) = (\frac{n(n+1)}{2})^2$

推推推

$$S(n)=\sum_{g=1}^{n}t(g)H(\left\lfloor\frac{n}{g}\right\rfloor)$$

其中 $t(n)=\mu(n)\cdot n^2, H(n)=\sum_{i=1}^{n}i\cdot f(\left\lfloor\frac{n}{i}\right\rfloor)$

通过预处理前 $n^{2/3}$ 项 H 可以轻易做到 $O(n^{2/3})$ 的复杂度，现在考虑 $t(n)$ 前缀和 $T(n)$。

筛！

$$\sum_{i=1}^{n}\sum_{d|i} t(i)(\frac{i}{d})^2 = \sum_{i=1}^{n} i^2 \sum_{d|i} \mu(d) = 1$$

$$\sum_{i=1}^{n}\sum_{d|i} t(i)(\frac{i}{d})^2 = \sum_{d=1}^{n} d^2 \sum_{i=1}^{\lfloor \frac{n}{d} \rfloor} t(i) = \sum_{d=1}^{n} d^2\, T(\left\lfloor \frac{n}{d} \right\rfloor)$$

这样就可以杜教筛了呀，问题解决，$O(n^{2/3})$。

再来看一道题

$$S(n)=\sum_{i=1}^{n}\sum_{j=1}^{n}d(ij)$$

$$n\le 10^{10}$$

推推推

$$S(n)=\sum_{i=1}^{n}\sum_{j=1}^{n}d(ij)$$
$$=\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{a|i}\sum_{b|j}[\gcd(a,b)=1]$$
$$=\sum_{d=1}^{n}\mu(d)\sum_{a=1}^{\lfloor\frac{n}{d}\rfloor}\left\lfloor\frac{n}{ad}\right\rfloor\sum_{b=1}^{\lfloor\frac{n}{d}\rfloor}\left\lfloor\frac{n}{bd}\right\rfloor$$

到这一步就很显然了，前后都可以通过预处理做到 $O(n^{2/3})$。

洲阁筛

然而除了杜老师搞出了杜教筛，洲阁也搞出了洲阁筛。

洲阁筛是一种在 $O(\frac{n^{3/4}}{\log n})$ 复杂度内求“大多数”积性函数前缀和的方法。

洲阁筛

设 $F(n)$ 为一个积性函数，n 的质因数分解为：$n = \prod_{i=1}^{k} p_i^{c_i}$，考虑这样描述一个积性函数：

- 当 n 为质数时，$n = p, F(p) = G(p)$
- 当 n 为质数幂时，即 $n = p^c$，且 $c > 1$，$F(p^c) = T(p^c)$
- 否组考虑积性，即 $F(n) = \prod_{i=1}^{k} F(p_i^{c_i})$

洲阁筛

比如对于两个常见函数：

- 欧拉函数 $\varphi(n)$： $G(p)=p-1, T(p^c)=(p-1)p^{c-1}$
- 莫比乌斯函数 $\mu(n)$： $G(p)=-1, T(p^c)=0$

更一般的， $G(p)$ 可以是一个与 p 相关的多项式， $T(p^c)$ 可以是一个与 p, c 相关的多项式，这种情况下一般很难用杜教筛做，现在我们考虑直接用积性来做。

洲阁筛

以这一类问题的一个简单形式作为例子。

定义一个欧拉函数的变种 $\varphi(n, d)$，$n = \prod_{i=1}^{k} p_i^{c_i}$，那么：

$$\varphi(n, d) = \prod_{i=1}^{k} (p_i^{c_i} + d)$$

特别的，定义 $\varphi(1, d) = 1$。

给定 n, d，求：

$$\sum_{i=1}^{n} \varphi(i, d)$$

答案对一个质数取模。

洲阁筛

考虑到 x 最多拥有一个 $>\sqrt{n}$ 的因子，根据积性有：

$$\sum_{i=1}^{n} F(i)=\sum_{\substack{x\le n\\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)\left(1+\sum_{\substack{\sqrt{n}<p\le\left\lfloor\frac{n}{x}\right\rfloor\\ p\text{为质数}}} G(p)\right)$$

显然我们可以根据 $\left\lfloor\frac{n}{x}\right\rfloor$ 将 F 分成 $O(\sqrt{n})$ 段。

设 $y=\left\lfloor\frac{n}{x}\right\rfloor$，那么我们要计算两部分：

$$\sum_{\substack{\sqrt{n}<p\le y\\ p\text{为质数}}} G(p) \qquad \sum_{\substack{\left\lfloor\frac{n}{x}\right\rfloor=y\\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)$$

计算 $G(p)$

如果 $G(p)$ 为多项式，可以对多项式每一项分别计算。

设不超过 $\sqrt{n}$ 的质数共 m 个，升序排列为 $p_1 \sim p_m$，设 $g_k[i][j]$ 表示 $[1,j]$ 范围内与前 i 个质数互质的所有数的 k 次幂之和，其中 $i=0$ 时可以插值 $O(k)$ 计算。

对于 $i>0$ 的情况，考虑 $[1,j]$ 范围内有多少含有 p_i 质因子且与前 $i-1$ 个质数互质的数，设 $l=\left\lfloor \frac{j}{p_i} \right\rfloor$，那么有：

$$g_k[i][j] = g_k[i-1][j] - p_i^k g_k[i-1][l]$$

递推最终得到的 $g_k[m][j]-1$ 即为 $[1,j]$ 内所有大于 $\sqrt{n}$ 的质数的 k 次幂和。

朴素实现

显然 $g_k[m][n]$ 的第二维状态数为 $O(\sqrt{n})$。

暴力的话，考虑到当 $y > \sqrt{n}$ 时，所有的 m 个质数都要被转移，而 $m = O(\frac{\sqrt{n}}{\log \sqrt{n}})$，因此：

$$\sum_{i=1}^{\lfloor \sqrt{n} \rfloor} O(\frac{\sqrt{n}}{\log \sqrt{n}}) \approx O(\frac{n}{\log n})$$

显然还是慢了点。

优化

当 $p_{i+1} > j$ 时，显然 $g_k[i][j] = 1$。
所以当 $p_i^2 > j \geq p_i$ 时，即 $l < p_i$ 时：

$$g_k[i-1][l] = 1$$

$$g_k[i][j] = g_k[i-1][j] - p_i^k$$

因此 $p_i^2 > j$ 时的计算是可以被省去的，之后再用的时候直接加上若干个质数的 k 次幂即可，所以对于每个 y，都只要转移不超过 $\sqrt{y}$ 的质数，复杂度可以估算为：

$$\sum_{i=1}^{\lfloor\sqrt{n}\rfloor} O(\frac{\sqrt{\frac{n}{i}}}{\log\sqrt{\frac{n}{i}}}) \approx O(\frac{n^{3/4}}{\log n})$$

计算 $F(x)$

考虑到 $y = \left\lfloor \frac{n}{x} \right\rfloor$ 对应了一段 x 的区间，可以计算一个前缀和的差，于是现在问题变成了计算 F 的前缀和，即：

$$\sum_{\substack{x \le L \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)$$

计算 $F(x)$

类似的，考虑 $f_k[i][j]$ 表示 $[1,j]$ 中只包含 $i\sim m$ 质数的数的 $F(x)$ 之和。

$$f_k[i][j]=f_k[i+1][j]+\sum_{c\geq 1}F(p_i^c)f_k[i+1][\left\lfloor\frac{j}{p_i^c}\right\rfloor]$$

这个暴力的复杂度和之前一样，也是 $O(\frac{n}{\log n})$ 的，太慢了。

故技重施

类似的，当 $p_i^2 > j$ 时，需要考虑的数是 $[p_i, \min(j, p_m)]$ 范围内的质数的 F 之和，也可以不做，到后面要用的时候再加上去，复杂度一样是 $O(\frac{n^{3/4}}{\log n})$。

SPOJ DIVCNT3

$$\sum_{i=1}^{n} d(i^3)$$

$$n \le 10^{11}, TL = 20s$$

SPOJ DIVCNT3

$$\sum_{i=1}^{n} d(i^3)$$

$$n \leq 10^{11}, TL = 20s$$

$F(p^c) = 3c + 1$，直接上洲阁筛。

容斥原理

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{i=1}^{n} |A_i| - \sum_{i<j} \left|A_i \bigcap A_j\right| + \sum_{i<j<k} \left|A_i \bigcap A_j \bigcap A_k\right| - \dots$$

二项式系数定义

$$\binom{n}{m} = \frac{n!}{m!\,(n-m)!}$$

组合意义：从 n 个不同物品中选 m 个物品的方案数。

显然的性质：

$$\sum_{i=0}^{n} \binom{n}{i} = 2^n$$

二项式定理

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^i y^{n-i}$$

一些可能有用的恒等式

$$\binom{n}{m} = \binom{n}{n-m}$$

$$k\binom{n}{k} = n\binom{n-1}{k-1}$$

$$\binom{n}{m} = \binom{n-1}{m-1} + \binom{n-1}{m}$$

$$\sum_{i=0}^{n}(-1)^i\binom{n}{i} = 0$$

一些可能有用的恒等式

$$\sum_{i=0}^{n} i \cdot \binom{n}{i} = n \cdot 2^{n-1}$$

$$\sum_{i=0}^{n} i^2 \cdot \binom{n}{i} = n(n+1)2^{n-2}$$

$$\sum_{i=0}^{n} \binom{n}{i}^2 = \binom{2n}{n}$$

$$\sum_{i=0}^{k} \binom{n+i}{i} = \binom{n+k+1}{k}$$

二项式反演

设存在两个数列满足 $a_n = \sum_{i=0}^{n} \binom{n}{i} b_i$，那么有：

$$b_n = \sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} a_i$$

证明

$$\begin{aligned}
a_n &= \sum_{i=0}^{n}\binom{n}{i}b_i \\
&= \sum_{i=0}^{n}\binom{n}{i}\sum_{j=0}^{i}(-1)^{i-j}\binom{i}{j}a_j \\
&= \sum_{i=0}^{n}a_i\sum_{j=i}^{n}\binom{n}{j}(-1)^{j-i}\binom{j}{i} \\
&= \sum_{i=0}^{n}a_i\sum_{j=i}^{n}(-1)^{j-i}\binom{n}{i}\binom{n-i}{j-i} \\
&= \sum_{i=0}^{n}a_i\binom{n}{i}\sum_{j=i}^{n}(-1)^{j-i}\binom{n-i}{j-i} = a_n
\end{aligned}$$

第二类斯特林数的反演

$$\left\{ \begin{matrix} n \\ m \end{matrix} \right\} = \frac{1}{m!} \sum_{i=0}^{m} (-1)^i \binom{m}{i} (m-i)^n$$

证明

先假设选出来的 m 个集合各不相同，枚举空集的个数容斥，最后再除掉 $m!$。

错排问题

求有多少个长度为 n 的排列 $\{a_i\}$，满足 $\forall 1 \le i \le n, a_i \neq i$。

错排问题

求有多少个长度为 n 的排列 $\{a_i\}$，满足 $\forall 1 \le i \le n, a_i \neq i$。

我相信大家都知道递推的做法，$f_n = (n-1)(f_{n-1} + f_{n-2})$，我们来思考一下反演的做法。

错排问题

设 f_i 表示有 i 个位置 $a_i \neq i$，那么有：

$$n! = \sum_{i=0}^{n} \binom{n}{i} f_i$$

二项式反演一下：

$$\begin{aligned} f_n &= \sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} i! \\ &= \sum_{i=0}^{n} (-1)^{n-i} \frac{n!}{(n-i)!} \\ &= n! \sum_{i=0}^{n} \frac{(-1)^i}{i!} \end{aligned}$$

球染色问题

有 n 个球排成一行，有 k 种颜色，给每个球染色并且要求相邻不同色，并且每种颜色必须使用的方案数。

球染色问题

有 n 个球排成一行，有 k 种颜色，给每个球染色并且要求相邻不同色，并且每种颜色必须使用的方案数。

如果没有每个颜色都使用的要求，那么方案数为 $k(k-1)^{n-1}$，设 f_k 表示答案，那么有：

$$k(k-1)^{n-1}=\sum_{i=0}^{k}\binom{k}{i}f_i$$

反演一波：

$$f_k=\sum_{i=0}^{k}(-1)^{k-i}\binom{k}{i}i(i-1)^{n-1}$$

这样我们就得出了答案。

群

设一个群 $<G,*>$ 表示其作用集合为 G，运算为 $*$，一个群 $<G,*>$ 被称为群，当且仅当满足：

- 封闭性：$\forall_{a,b\in G},(a*b)\in G$
- 结合律：$\forall_{a,b,c\in G},a*(b*c)=(a*b)*c$
- 单位元：$\exists_{e\in G}\forall_{a\in G}a*e=e*a=a$
- 逆元：$\forall_{a\in G}\exists_{b\in G}a*b=e$

Burnside 引理

设 $< G, * >$ 是 $[1, n]$ 上的置换群，记 $F(p)$ 表示置换 p 的不动点数，那么等价类数为：

$$\frac{\sum_{p \in G} F(p)}{|G|}$$

将置换 p 表示成若干个轮换的积，那么每个轮换的方案必须相同。换句话说就是能使得每个轮换中的元素相同的方案。

Pólya 定理

设 $< G, * >$ 是 $[1, n]$ 上的置换群，现在有 m 种颜色，记 $c(p)$ 表示置换 p 的轮换数，那么等价类数为：

$$\frac{\sum_{p \in G} m^{c(p)}}{|G|}$$

其实就是 Burnside 的一种特殊形式。

一道练手题

项链由 n 个珠子构成，有 m 种珠子，每种都有无数个。珠子间有 k 对关系，每对关系 (a,b) 表示 a,b 不能相邻。求本质不同的项链个数，答案对 9973(一个质数) 取模。满足 $n \le 10^9, \gcd(n, 9973) = 1, m \le 10$。

一道练手题

n 太大了，很显然不能枚举所有置换。

对于移位 p 的置换，轮换数就是 $\gcd(p,n)$，可以看出答案显然只和轮换数有关。

考虑轮换数为 d 的置换的数量是等于 $\varphi(\frac{n}{d})$，因此我们枚举 n 的约数 d，算出这个 d 下的答案再乘上 $\varphi(\frac{n}{d})$ 即可，算一个轮换数下的答案用矩阵乘法即可，复杂度 $O(d(n)(\sqrt{n}+m^3\log n))$。

一道 sgu 题

染色图是无向完全图，且每条边可被染成 M 种颜色中的一种。两个图是同构的，当且仅当可以改变一个图中的顶点的编号，使得两个染色图完全相同。问 N 个顶点，M 种颜色，本质不同的染色图的个数，答案对一个质数 P 取模。

$N \leq 53, M \leq 1000, N < P \leq 10^9, TL = 10s$

一道 sgu 题

显然置换个数有 $N!$ 个，不可能直接枚举，考虑到每个置换是由若干轮换乘起来的，那么我们可以枚举 N 的整数分拆，接下来的任务分成两部分，计算有多少置换满足这个整数分拆，计算这个整数分拆下的答案。

计算置换数

考虑枚举的拆分为 $x_1, x_2, \cdots, x_r, \sum_{i=1}^{r} x_i = N$，考虑将 N 个元素填到 r 个环中去，那么方案数是 $N!$，一个环转一下还是一样的，所以要除掉 $\prod_{i=1}^{r} x_i$，同时环长相同的环之间顺序是不重要的，设 c_i 表示长度为 i 的环的数量，那么最后得出的方案数就是：

$$\frac{N!}{\prod_{i=1}^{r} x_i \prod_{i=1}^{N} c_i!}$$

计算方案数

点置换 $i \to p_i$，那么边置换 $(i,j) \to (p_i,p_j)$，考虑将点置换变成边置换。

对于两点都在一个环 x_i 内的边，边置换环的数量很显然是 $\lfloor \frac{x_i}{2} \rfloor$，对于两点在不同环 x_i,x_j 上的边，边置换环数显然为 $\frac{x_ix_j}{\mathrm{lcm}(x_i,x_j)} = \gcd(x_i,x_j)$。

那么把所有边置换环的数量累加起来记为 $s(p)$，这里的方案数就是 $M^{s(p)}$，p 表示此时的点置换。

有标号无向图

令 $G_{p,q}$ 表示 p 个点 q 条边的有标号无向图的个数，G_p 表示 p 个点的有标号无向图的个数，那么：

$$G_{p,q} = \binom{\binom{p}{2}}{q}$$

$$G_p = 2^{\binom{p}{2}}$$

有标号无向连通图

有标号无向连通图

令 C_p 表示 p 个点的有标号无向连通图的个数，那么：

$$C_p = G_p - \sum_{i=0}^{p-2} \binom{p-1}{i} C_{i+1} G_{p-i-1}$$

有标号有向无环图

求 n 个点的 DAG 个数，答案对一个 998244353 取模。

- version 1 $n \le 5000$
- version 2 $n \le 100000$
- version 3 要求弱连通，$n \le 5000$
- version 4 要求弱连通，$n \le 100000$

version 1

version 1

考虑 dp，设 f_n 表示 n 个点的 DAG 数量，根据容斥原理有：

$$f_n = \sum_{i=1}^{n} (-1)^{i-1} \binom{n}{i} 2^{i(n-i)} f_{n-i}$$

复杂度 $O(n^2)$。

version 2

version 2

$$f_n=\sum_{i=1}^{n}(-1)^{i-1}\frac{n!}{i!\,(n-i)!}2^{i(n-i)}f_{n-i}$$
$$=n!\cdot 2^{\frac{n^2}{2}}\sum_{i=1}^{n}\frac{(-1)^{i-1}}{i!\cdot 2^{\frac{i^2}{2}}}\frac{f_{n-i}}{(n-i)!\cdot 2^{\frac{(n-i)^2}{2}}}$$

恰巧 2 在模 998244353 下存在二次剩余，这样的话我们可以分治 FFT 计算答案，$O(n\log^2 n)$。

version 2 faster

version 2 faster

观察一下这个式子：

$$f_n = n! \cdot 2^{\frac{n^2}{2}} \sum_{i=1}^{n} \frac{(-1)^{i-1}}{i! \cdot 2^{\frac{i^2}{2}}} \frac{f_{n-i}}{(n-i)! \cdot 2^{\frac{(n-i)^2}{2}}}$$

可以发现我们如果设 $g_n = \frac{f_n}{n! \cdot 2^{\frac{n^2}{2}}}$，那么有：

$$g_n = \sum_{i=1}^{n} \frac{(-1)^{i-1}}{i! \cdot 2^{\frac{i^2}{2}}} g_{n-i}$$

version 2 faster

设 $h_n = \frac{(-1)^{n-1}}{n!\cdot 2^{\frac{n^2}{2}}}$，那么有：

$$g_n = \sum_{i=1}^{n} h_i g_{n-i} \to \sum_{i=0}^{n} h_i g_{n-i} = 0, n > 0$$

$$h \cdot g = -1$$

多项式逆元即可，$O(n \log n)$。

version 3

设 g_n 表示联通的 DAG 数，f_n 表示 DAG 数，那么：

$$g_n = f_n - \sum_{i=0}^{n-2} \binom{n-1}{i} g_{i+1} f_{n-i-1}$$

$O(n^2)$。

version 4

$$f = e^g, g = \ln f$$

多项式 ln 即可，$O(n \log n)$。

有标号无根树个数

有标号无根树个数

n 个点的有标号无根树的个数是 n^{n-2} 个。

证明？

有标号无根树个数

n 个点的有标号无根树的个数是 n^{n-2} 个。

证明？

Prufer 序列 or Matrix-tree 定理。

Prufer 序列

一种有标号无根树的序列，n 个点的树的序列长度为 $n-2$。

Prufer 序列

一种有标号无根树的序列，n 个点的树的序列长度为 $n-2$。

- 树 $\rightarrow$ 序列：迭代，每次找到编号最小的叶子节点，将与其相邻的点的编号加入序列，然后删掉这个叶子节点，直到最后剩下 2 个点时结束。

Prufer 序列

一种有标号无根树的序列，n 个点的树的序列长度为 $n-2$。

- 树 $\to$ 序列：迭代，每次找到编号最小的叶子节点，将与其相邻的点的编号加入序列，然后删掉这个叶子节点，直到最后剩下 2 个点时结束。
- 序列 $\to$ 树：考虑维护一个集合 S 表示当前叶子节点的集合，假设当前考虑到序列第 i 位，将 S 中编号最小的点和 p_i 连边，然后将 S 中编号最小的点从 S 中删除，最后将剩下的两个点连一条边即可。

Prufer 序列

一种有标号无根树的序列，n 个点的树的序列长度为 $n-2$。

- 树 $\rightarrow$ 序列：迭代，每次找到编号最小的叶子节点，将与其相邻的点的编号加入序列，然后删掉这个叶子节点，直到最后剩下 2 个点时结束。
- 序列 $\rightarrow$ 树：考虑维护一个集合 S 表示当前叶子节点的集合，假设当前考虑到序列第 i 位，将 S 中编号最小的点和 p_i 连边，然后将 S 中编号最小的点从 S 中删除，最后将剩下的两个点连一条边即可。

可以发现有标号无根树和 Prufer 序列是一一对应的。

因此 Prufer 序列的个数 $=$ 有标号无根树的个数 $= n^{n-2}$。

一道 HNOI 题

自从明明学了树的结构，就对奇怪的树产生了兴趣。给出标号为 1 到 n 的点，以及某些点最终的度数，允许在任意两点间连线，可产生多少棵度数满足要求的树？

$n \leq 1000$

注意没说要取模。

一道 HNOI 题

设有 k 个度数要求，分别是 $d_1, d_2, \cdots, d_k$，那么方案数：

$$\binom{n-2}{\sum_{i=1}^{k}(d_i-1)} \frac{(\sum_{i=1}^{k}(d_i-1))!}{\prod_{i=1}^{k}(d_i-1)!}(n-k)^{n-2-\sum_{i=1}^{k}(d_i-1)}$$

高精度差评。

Matrix-tree 定理

对一个有标号无向图定义基尔霍夫矩阵 G，满足：

对于 $i \neq j$，$G_{i,j} = -1$ 当且仅当 i, j 之间有一条边。

对于 $i = j$，$G_{i,i}$ 表示 i 的度数。

换句话说，就是度数矩阵减去邻接矩阵。

$\forall i \in [1, n]$，去掉 G 的第 i 行第 i 列后矩阵的行列式就是这个图的生成树个数，复杂度 $O(n^3)$，证明略 (其实是不会)。

小扩展

给定一个有编号无向图，图上每条边是红色或蓝色，求恰有 k 条红边的生成树个数。

$n \leq 50$

小扩展

给定一个有编号无向图，图上每条边是红色或蓝色，求恰有 k 条红边的生成树个数。

$n \leq 50$

将红边视为 x，蓝边视为 1，那么求出来的行列式的 x^k 的系数即是答案，考虑行列式的多项式是 $O(n)$ 的，插值即可，$O(n^4)$。

一道 UR 题

给定 n 个点的图，对于每个生成的基环外向数定义权值 2^w，其中 w 为度数大于 1 的点的数量，求所有基环外向树的权值和。

$n \leq 16$

一道 UR 题

设 $f(V)$ 表示基环外向树的集合为 V 的答案，那么有：

$$f(V) = \sum_{T \subset V} (-1)^{|T|} \cdot 2^{|V|-|T|} \cdot ways(T, V-T) \cdot cnt(V-T)$$

这里考虑用叶子节点集合来容斥，其中 $ways(T, V-T)$ 表示每个 T 中的点在 $V-T$ 中相邻的点的个数的乘积，也就是把 T 的结点当做叶子节点加入 $V-T$ 中构成基环外向树的方案数，$cnt(V-T)$ 表示 $V-T$ 的基环外向树的个数。

ways 可以用 dfs 做到 $O(3^n)$，考虑怎么算 cnt。

一道 UR 题

显然同样有一个容斥的式子：

$$cnt(V) = cir(V) + \sum_{T\subset V, T\neq\phi} (-1)^{|T|-1} \cdot ways(T, V-T) \cdot cnt(V-T)$$

其中 $cir(V)$ 表示 V 构成一个环的方案数，考虑 dp，设 $g(V,p)$ 表示当前集合为 V，链上当前点为 p，链的开头为链上编号最小的点，方程很显然。记得要把方案数除以 2，因为一个环可以被两个方向都算一遍，这里是 $O(2^n n^2)$ 的，因此最终复杂度就是 $O(2^n n^2 + 3^n)$。