数学

wwwwodddd

2019 年 1 月 29 日

目录

1 自我介绍

毕克
2010 ~ 2013，石家庄二中。
NOI 2012 金牌。
2013 ~ 2017，清华大学。
5 次区域赛/EC-FINAL 金牌。
没去过 IOI，也没去过 ACM World Final。
三大爱好：比赛，出题，讲课。
My vegetable exploded up.
~~QQ:751723392, Email:wwwwodddd@gmail.com~~

2 基础知识

2.1 取模

因为取模具有很好的性质，比如

$$(a+b) \bmod p = ((a \bmod p) + (b \bmod p)) \bmod p$$

$$(a-b) \bmod p = ((a \bmod p) - (b \bmod p)) \bmod p$$

$$(a \times b) \bmod p = ((a \bmod p) \times (b \bmod p)) \bmod p$$

对于除法，还可以乘法逆元。
所以如果最终的答案比较大，通过要求取模是很常见的做法。

2.2 快速幂

经常需要计算比如 $a^n \bmod p$ 的结果。

结合律

有结合律的都可以，比如说加法，乘法，矩阵乘法，排列。
[NOI2012] 随机数生成器

十进制

不仅可以二进制快速幂，如果输入的数字是高精度，并不需要把数字转化为二进制，可以直接十进制快速幂。
[NOI2013] 矩阵游戏

2.3 最大公约数

欧几里得算法。
高精度为了避免除法，二进制分奇偶讨论。

2.4 最小公倍数

利用 $\operatorname{lcm}(x,y) = xy/\gcd(x,y)$

2.5 （扩展）欧几里得算法

已知 a, b。

求 x, y 使得 $ax + by = \gcd(a, b)$。

2.6 其他欧几里得算法

《欧几里得算法的应用》写的很全面了。

其中比较值得注意的是用欧几里得算法求行列式的值。

还有一些利用辗转相除的性质解决一些题目的例子。

2.7 费马小定理

对于质数 p，和与质数 p 互质的 a，满足

$$a^{p-1} \bmod p = 1$$

2.8 欧拉定理

对于一般的数 p，和与 p 互质的 a，满足

$$a^{\varphi(p)} \bmod p = 1$$

2.9 原根

对于一般的数 p，和与 p 互质的 a，求出最小的 d 满足

$$a^{d} \bmod p = 1$$

d 一定是 $\varphi(p)$ 的约数，枚举约数即可。（事实上枚举所有质因子一个个除即可。）

如果 $d = \varphi(p)$ 那么 a 是 p 的原根。

2.10 离散对数/Baby Step, Gaint Step

已知 a, b, p 求 x 使得 $b^x = a \pmod p$。

并没有特别有效的做法，一个时间复杂度为 $\sqrt{p}$ 的做法是这样的：

设 $t = \sqrt{p}, x = ct - d(0 \le d < t)$，原式变为 $b^{ct-d} = a \pmod p$。

进一步 $(b^t)^c = ab^d \pmod p$。

注意到右侧的 d 只有 t 个取值，所以最终只有 t 个结果，可以预处理出这 t 个结果，枚举左边的 c，检查是否是答案。

因为如果有解，一定小于 p（实际上是小于 $\varphi(p)$）所以时间复杂度是 $O(\sqrt{p})$。

这个方法要求 b 和 p 互质。

2.11 乘法逆元

如果 $ab \bmod p = 1$，那么 a 是 b 的乘法逆元，b 也是 a 的乘法逆元。
根据费马小定理

$$a^{p-2} \times a \bmod p = 1$$

所以 a 的逆元是 $a^{p-2} \bmod p$ 可以用快速幂计算。
对于模一般数字，用扩展欧几里得。
并不用欧拉定理和快速幂，因为求 $\varphi(n)$ 的时间复杂度 $O(\sqrt{n})$ 而不是 $O(\log n)$。

2.12 中国剩余定理

解模方程，如果取模的数字都互质，那么可以用。
如果不互质，那么用扩展欧几里得更好。
对于每个质数 p_i，都要找到一个数字 m_i 满足 $m_i \bmod p_i = 1, m_i \bmod p_j = 0 (j \neq i)$。
最后 $\sum_i r_i m_i$ 就是答案。

2.13 long long 乘以 long long 模 long long

1. 类似快速幂的乘法。
2. long long 越界，double 近似算商。
3. 内联汇编（OI 可能不让用）
4. 如果数字不大，比如小于 2^{42}，那么可以分为两段各 21 位，分别进行乘法。
5. 分解质因数，最后中国剩余定理合并。

2.14 约数个数，约数和

分解质因数（唯一分解定理）

$$n = \prod_{i=1}^{k} p_i^{a_i}$$

n 的约数个数

$$\prod_{i=1}^{k} (a_i + 1)$$

n 的约数和

$$\prod_{i=1}^{k} \sum_{j=0}^{a_i} p_i^j$$

2.15 判断质数

- 暴力 $O(\sqrt{n})$。
- Miller Robin $O(\log n)$

2.16 分解质因数

- 暴力 $O(\sqrt{n})$。
- Pollard Rho $O(n^{1/4})$

2.17 二次剩余

3 poj 1006

3.1 题目背景

poj 1006 Biorhythms

3.2 题目大意

找到最小的正整数 x 满足

$$x = p \pmod{23}$$

$$x = e \pmod{28}$$

$$x = i \pmod{33}$$

$$x > d$$

输出 $x - d$。

3.3 题目分析

```
p*5544+e*14421+i*1288-d
```

然后将答案调到 1 到 21252 即可。

4 Luogu P1495

4.1 题目背景

P1495 曹冲养猪

4.2 题目大意

这题出得非常垃圾。
答案在 long long 之内。

4.3 题目解法

注意到每个 a_i 很小，所以可以不用逆元，直接暴力即可。
我的代码

5 贾志鹏线性筛

链接

1. $O(n \log \log n)$ 筛法
2. $O(n)$ 筛法
3. 利用筛法计算最小（大）质因子
4. （完全）积性函数
5. 利用筛法计算积性函数的值
6. 常见完全积性函数：$f(x) = x^k$。
7. 常见积性函数：φ 欧拉函数，μ 莫比乌斯函数。
8. 线性筛法求逆元 $inv(x) = inv(p \bmod x) \times (p - p/i)$。
9. 预处理阶乘的逆元求组合数。

另一个我觉得写的不错的课件莫比乌斯反演

6 莫比乌斯反演

核心公式

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

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

7 bzoj 1257

7.1 题目背景

[CQOI2007] 余数之和

7.2 题目大意

输入 n, k，计算

$$\sum_{1 \le i \le n} k \bmod i$$

7.3 题目分析

核心思想就是 $\lfloor \frac{k}{i} \rfloor (1 \le i \le n)$ 只会有 $\sqrt{k}$ 个结果。
再结合上 $k \bmod i = k - i \times \lfloor \frac{k}{i} \rfloor$。
对于 $\lfloor \frac{k}{i} \rfloor$ 相同的一起处理即可。

8 bzoj 2956

8.1 题目背景

模积和

送我退役的题（之一）

8.2 题目大意

输入 n, m，求

$$\sum_{1\le i\le n, 1\le j\le m, i\ne j} (n \bmod i) \times (m \bmod j)$$

8.3 题目分析

样例出的非常垃圾。大概就是要看到 $i \ne j$。

然后就是全部减去相等的部分。

相等的部分就是两个等差数列相乘。

9 bzoj 2154

9.1 题目背景

Crash 的数字表格

（贾志鹏太强啦）

9.2 题目大意

输入 n, m，求

$$\sum_{1\le i\le n}\sum_{1\le j\le m} \mathrm{lcm}(i, j)$$

答案对质数取模。

9.3 题目分析

就是贾志鹏论文上的题目。

参考题解

大概就是经过一系列推倒之后得到了 (11) 那个表达式。

然后需要对每个 $T(1 \le T \le n)$ 计算 $\sum_{i|T} \mu(i)i$。

这是一个积性函数，具体来说就是脑补质因子 p 一个一个增加。

如果增加的出现过，答案不变，如果增加的没出现过，那么 $f(pk) = (1-p)f(k)$。

10 bzoj 2693

10.1 题目背景

JZPTAB
（贾志鹏太强啦）

10.2 题目大意

输入 n, m，求

$$\sum_{1\leq i\leq n}\sum_{1\leq j\leq m}\operatorname{lcm}(i,j)$$

答案对质数取模。
与之前的区别是多组询问。

10.3 题目分析

上一个题目再加一个分块就行了。
参考代码
注意这份代码是把 $D\sum_{d|D} d\mu(d)$ 一起计算的，与上一份略有不同。

11 bzoj 2820

11.1 题目背景

YY 的 GCD

11.2 题目大意

输入 n, m，求
问有多少对 x, y 满足 $1 \leq x \leq n, 1 \leq y \leq m$ 并且 $\gcd(i, j)$ 为质数。

11.3 题目分析

回想之前做过的题目，大概我们会做是否互质（莫比乌斯函数），或者是直接求和（欧拉函数）
那么对于这次我们可以构造一个新函数 $t(i)$。

$$P(n) = [n\text{是质数}] = \sum_{i|n} t(i)$$

根据莫比乌斯反演

$$t(n) = \sum_{i|n} P(i)\mu\left(\frac{n}{i}\right)$$

相当于直接枚举 n 的所有质数约数，就可以预处理出 $t(n)$ 的值。
这样 1 到 n 所有数字的质数约数个数大概是 $\frac{n}{2}+\frac{n}{3}+\frac{n}{5}+\cdots = n\log\log n$。
参考题解

12 杜教筛

12.1 简单介绍

设

$$S(n) = \sum_{1 \le i \le n} \mu(i)$$

回顾公式

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

对 n 求和

$$\sum_{1 \le i \le n} \sum_{d|i} \mu(d) = 1$$

换一种求和。

$$\sum_{1 \le i \le n} \sum_{1 \le d \le \lfloor n/i \rfloor} \mu(d) = 1$$

拆成两部分

$$\sum_{1 \le i \le n} \mu(i) = 1 - \sum_{2 \le i \le n} \sum_{1 \le d \le \lfloor n/i \rfloor} \mu(d)$$

注意到右边遇到的都是子问题。

$$S(n) = 1 - \sum_{2 \le i \le n} S\left(\lfloor \frac{n}{i} \rfloor\right)$$

注意到连除有结合律 $\lfloor \frac{\lfloor \frac{a}{b} \rfloor}{c} \rfloor = \lfloor \frac{a}{bc} \rfloor$

12.2 时间复杂度

本质上只需要计算 $S(\lfloor \frac{n}{i} \rfloor)(1 \le i \le n)$。

大家都知道 $\lfloor \frac{n}{i} \rfloor$ 本质不同的结果只有 $2\sqrt{n}$ 个，这 $2\sqrt{n}$ 个结果可以分为两部分。

分别为当 $1 \le i \le \sqrt{n}$ 时的商 $\lfloor \frac{n}{i} \rfloor$ 和当 $\sqrt{n} \le i \le n$ 的商 $1, 2, 3, \ldots, \sqrt{n}$。

在已知 $S(\lfloor \frac{n}{i} \rfloor)(2 \le i \le n)$ 的情况下，计算 $S(n)$ 的复杂度为 $O(\sqrt{n})$。

所以总时间复杂度为

$$\sum_{1 \le i \le \sqrt{n}} \sqrt{\frac{n}{i}} + \sum_{1 \le i \le \sqrt{n}} \sqrt{i}$$

利用把求和换成积分的技巧，原式等于

$$\int_1^{\sqrt{n}} \sqrt{\frac{n}{x}} \mathrm{d}x + \int_1^{\sqrt{n}} \sqrt{x} \mathrm{d}x$$

不定积分的结果为

$$\int \sqrt{\frac{n}{x}} \mathrm{d}x = 2\sqrt{nx}$$

$$\int \sqrt{x} \mathrm{d}x = \frac{2}{3} x^{\frac{3}{2}}$$

带入 $x=\sqrt{n}$ 可以得到原式为 $O(n^{\frac{3}{4}})$，详细计算过程为

$$\int_1^{\sqrt{n}}\sqrt{\frac{n}{x}}\mathrm{d}x+\int_1^{\sqrt{n}}\sqrt{x}\mathrm{d}x=(2\sqrt{n^{\frac{3}{2}}}-2\sqrt{n})+(\frac{2}{3}(\sqrt{n})^{\frac{3}{2}}-\frac{2}{3})\approx\frac{8}{3}n^{\frac{3}{4}}=O(n^{\frac{3}{4}})$$

如果希望用线性筛进一步优化，注意到 $1,2,3,\ldots,\sqrt{n}$ 必然可以被从 $O(n^{\frac{3}{4}})$ 优化到 $O(\sqrt{n})$。
但仅做此优化，并不解决问题，因为第一部分的复杂度还是 $O(n^{\frac{3}{4}})$。
引入参数 B，假设 $1,2,3,\ldots,\frac{n}{B}$ 用线性筛，其余部分用原方法，总复杂度为

$$\sum_{1\le i\le B}\sqrt{\frac{n}{i}}+\frac{n}{B}=2\sqrt{nB}+\frac{n}{B}$$

当 $B=O(n^{\frac{1}{3}})$ 时，两部分相等，取到最小值，时间复杂度为 $O(n^{\frac{2}{3}})$。
因此常见的数据范围是 10^{10} 或者 10^{11}。

12.3 欧拉函数

同样的技巧可以用在欧拉函数上。

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

只是右侧求和要换成 $n(n+1)/2$。

$$\sum_{1\le i\le n}\mu(i)=\frac{n(n+1)}{2}-\sum_{2\le i\le n}\sum_{1\le d\le\lfloor n/i\rfloor}\mu(d)$$

13 51nod 1239

13.1 题目背景

51nod 1239 欧拉函数之和

13.2 题目大意

输入一个 $n(1\le n\le 10^{10})$，求

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

13.3 参考代码

参考代码

14 51nod 1239

14.1 题目背景

51nod 1244 莫比乌斯函数之和

14.2 题目大意

输入一个 $A, B(1 \leq A \leq B \leq 10^{10})$，求

$$\sum_{i=A}^{B} \mu(i)$$

14.3 参考代码

参考代码

15 ProjectEuler 625

15.1 题目背景

PE 625 Gcd sum

15.2 题目大意

输入 n，求

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

结果模 998244353。

15.3 题目分析

直接杜教筛。

16 ProjectEuler 643

16.1 题目背景

PE 643 2-Friendly

16.2 题目大意

输入 n，求有多少 $p, q(1 \leq p < q \leq n)$ 满足 $\gcd(p, q) = 2^t$。
结果模 1000000007。

16.3 题目分析

直接杜教筛。

17 bzoj 4916

17.1 题目背景

神犇和蒟蒻

17.2 题目大意

输入 n，求 $\sum_{i=1}^{n}\mu(i^2)$ 和 $\sum_{i=1}^{n}\varphi(i^2)$。

17.3 题目分析

杜教筛的简单应用，第一个一定是 1。

第二个所求即是

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

考虑平方和

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

不再枚举 i 了，枚举 $x=\frac{i}{d}$

$$\sum_{i=1}^{n}\sum_{d|i}\varphi(d)\times d\times\frac{i}{d}=\sum_{x=1}^{n}x\sum_{d=1}^{\lfloor\frac{n}{x}\rfloor}\varphi(d)\times d=\sum_{x=1}^{n}xS(\lfloor\frac{n}{x}\rfloor)$$

所以就有

$$\frac{n(n+1)(2n+1)}{6}=\sum_{x=1}^{n}xS(\lfloor\frac{n}{x}\rfloor)$$

参考代码

18 bzoj 4176

18.1 题目背景

神犇和蒟蒻

18.2 题目大意

输入 n，求

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

其中 $d(ij)$ 表示 ij 约数的个数。

18.3 题目分析

首先你需要知道这样一个处理约数个数的结论

$$d(nm) = \sum_{i|n}\sum_{j|m}[\gcd(i,j)=1]$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k|i}\sum_{l|j}\sum_{d|\gcd(i,j)}\mu(d)$$

设

$$f(n) = \sum_{i=1}^{n}\lfloor\frac{n}{i}\rfloor$$

$$\sum_{d=1}^{n}\mu(d)f(\lfloor\frac{n}{d}\rfloor)^2$$

$\lfloor\frac{n}{d}\rfloor$ 只有很少的不同的值，对于对应位置，询问 $\mu(d)$ 的前缀和即可。

参考题解

19 浮点数

19.1 简介

与定点数相对，存储小数，最容易想到的方法就是乘以 10 的次幂，变成整数之后处理。

可以认为浮点数就是二进制的科学计数法。

如果关心具体每个数字怎么存，所有异常如何处理，可以参考IEEE 754

19.2 存储

float 占 4 字节，共 32 位，其中 1 位是符号位，8 位是指数位，23 位是底数位。

double 占 8 字节，共 64 位，其中 1 位是符号位，11 位是指数位，52 位是底数位。

long double 占 10 字节，共 80 位，其中 1 位是符号位，15 位是指数位，1 为是底数的整数部分，63 位是底数位。

19.3 例子一

输出 $0.1, 0.2, 0.3, 0.1 + 0.2$。

可以发现 $0.1 + 0.2 \neq 0.3$，这是因为在二进制下，这些数字都是无限循环小数，存在浮点数中都有误差。

19.4 例子二

输出 2^{53} 次方附近的整数。

可以注意到对于小于 2^{53} 的整数，double 类型均可以精确存储，大于 2^{53} 则开始有误差。

换句话说，只要 int 或 unsigned 类型可以存储的整数，double 类型均可以精确存储。

可以注意到对于小于 2^{64} 的整数，double 类型均可以精确存储，大于 2^{64} 则开始有误差。

换句话说，只要 long long 或 unsigned long long 类型可以存储的整数，long double 类型均可以精确存储。

19.5 例子三

这个例子和计算机系统有关系，在比较老的 Windows XP 上可能会失败。

2 的次幂可以精确输出。

0.5 的次幂可以精确输出，动态控制输出精度。

19.6 舍入

计算机中一般是四舍六入五留双，也就是当小数部分严格小于一半时舍去，严格大于一半时进位，严格等于一半时，由倒数第二位决定，舍入之后保证倒数第二位是偶数。

19.7 误差

参考 NOIP2017 奶酪。

19.8 抖动

在部分题目中为了防止三点共线之类的事情发生，需要加入随机误差。

20 概率期望

20.1 概型

- 古典概型，等概率的基本事件。
- 几何概型，按面积体积等随机。

值得注意的是在几何概型中，存在会发生，但是发生概率为 0 的事件。

20.2 独立事件

A 和 B 独立，当且仅当 $P(AB) = P(A)P(B)$。

A, B, C 独立，当且仅当两两独立，且 $P(ABC) = P(A)P(B)P(C)$。

20.3 期望

期望的定义：

$$\mathbb{E}[x] = \sum_i i \Pr[x = i]$$

期望是线性的：

$$\mathbb{E}[ax + b] = a\mathbb{E}[x] + b$$

$$\mathbb{E}[x+y]=\mathbb{E}[x]+\mathbb{E}[y]$$

乘法只有在 x 和 y 独立的情况下。

$$\mathbb{E}[xy]=\mathbb{E}[x]\mathbb{E}[y]$$

20.4 方差

方差的定义:

$$\mathrm{Var}(x)=\mathbb{E}[(x-\mathbb{E}[x])^2]$$

方差的重要性质:

$$\mathrm{Var}(x)=\mathbb{E}[x^2]-\mathbb{E}[x]^2$$

20.5 条件概率

$$\Pr[B|A]=\Pr[A\wedge B]/\Pr[A]$$

20.6 羊和车

有 3 个门，2 个后面是羊，1 个后面是车。

你可以选一个门，得到门后面的东西。但是你不知道门后面是什么。

这时你随机选了一个，然后主持人打开了一个门，后面是羊，问你改不改变自己的选择。

问题是改变是否可以提高门后面是车的概率?

应该改变，不改变是车的概率是 $\frac{1}{3}$，改变是车的概率 $\frac{2}{3}$。

门后面是车概率提高了，如果不便于理解，可以想象有 $n=10^9$ 个门，其中一个后面是车，然后主持人打开了 $n-2$ 个。

当然，改变的前提是你想要一个车而不是羊。

20.7 酒鬼

一个酒鬼每天 0.3 的概率去 A 酒吧，0.3 的概率去 B 酒吧，0.3 的概率去 C 酒吧，0.1 的概率在家。

警察想找到这个酒鬼，去看了 A,B 酒吧都没发现，问他在 C 酒吧的概率是多少?

答案

概率是 0.75。

20.8 生日问题一

如果认为每天出生概率相同。(虽然由于政策，季节，并不相同。)

并且认为任意两个人生日相互独立。

并且不考虑闰年。

选多少个人，有两个人生日相同的概率就超过了 0.5?

答案

23 个人。

$$\frac{365}{365} \times \frac{364}{365} \times \frac{363}{365} \times \cdots$$

20.9 生日问题二

如果认为每天出生概率相同。（虽然由于政策，季节，并不相同。）

并且认为任意两个人生日相互独立。

并且不考虑闰年。

选多少个人，有一个人与你生日相同的概率就超过了 0.5?

答案

$$\left(\frac{364}{365}\right)^n = 0.5$$

用对数（或者二分）算。$n = 253$ 时左侧略小于 0.5 满足条件。

20.10 生日问题三

如果认为每天出生概率相同。（虽然由于政策，季节，并不相同。）

并且认为任意两个人生日相互独立。

并且不考虑闰年。

你是一个良心老板，要雇人打工，你要雇佣 n 个人。

在一年中，对于一天来说，如果他是任意一个人的生日，那么这天所有人都放假。

作为一个良（黑）心老板，你当然想最大化人数乘以工作日，你希望知道期望的最大值是多大。

答案

注意期望是可以相加的，你只需要关注一天（比如第一天）有多少人能上班。

你并不需要以一年为周期考虑这个问题。

每个人会导致上班的概率乘以 $(1 - \frac{1}{365})$。

n 个人上班的话，收益是

$$n(1 - \frac{1}{365})^n$$

你可以通过打表，二分，求导，或者是解不等式来处理这个问题。

可以发现 $n = 364$ 和 $n = 365$ 同时是最大值。

20.11 两个孩子

已知两个孩子中其中一个是女孩。求另一个是男孩的概率。

我们默许男孩和女孩的概率均为 50%。

我们默许两人的性别相互独立。

答案是 $\frac{2}{3}$，因为题目中给的条件是，第一个是女孩或者第二个是女孩，这个条件成立。

附加题：如果其中一个是星期日出生的女孩。求另一个是男孩的概率。

答案是 $\frac{14}{27}$。

结论：随着对一个人了解的深入，另一个人的状态会越来越趋向平均。

20.12 猜数字

甲选择两个不同的实数 x_0, x_1 满足 $0 \le x_0, x_1 \le 1$。

乙选择其中一个 i，并且得知 x_i 具体是多少。

乙需要猜测另一个 x_{1-i} 和 x_i 的大小关系。

找一个乙胜率大于 50% 的策略。

换句话说，这个世界上有很多个甲。有的甲固定选 0.3 和 0.6；有的甲以 0.5 的概率选 0.1 和 0.2，以 0.5 的概率选 0.8 和 0.9；有的甲从 0 到 1 之间均匀随机选取两个数字。

你作为乙，要求你的策略，无论面对哪个甲，胜率都必须大于 50%。

策略：均匀随机一个 y，如果选到的 x_i 大于 y，就猜 x_i 较大。，如果选到的 x_i 小于 y，就猜 x_i 较小。

这样如果随机的 y 在 x_0, x_1 之间必胜，否则胜率 50%。

20.13 最擅长的英雄

B 君高中，刚开始去机房学 OI，只是为了打 Dota，B 君只打 AI。

B 君每次会随机选择一个英雄，你可以认为是从 0 到 1 中随机选择一个实数。

选择的数值越大，表示 B 君操作的越熟练，B 君可以和 AI 队友换英雄。

所以问从 0 到 1 中随机 k 个数字，最大值的期望是多少?

答案

$$\int_0^1 kx^k \mathrm{d}x = \frac{k}{k+1}$$

事实上这个结论可以推广，将他们从小到大排序之后，第 i 个数字期望是 $\frac{i}{k+1}$。

地震后的幻想乡

20.14 立起来的硬币

姚班招生题目 2013 物理第三题

将一个硬币抛起来如果立起来的可能性是 $\frac{1}{3}$ 求硬币与厚度的关系?

答案

Matrix 67 的解答
这个题有人跟我说，这么算是不对的。
因为抛起来是一个三维的随机，而不是二维的。
二维，直径是厚度的 $\sqrt{3}$ 倍。
三维，直径是厚度的 $\sqrt{8} = 2\sqrt{2}$ 倍。

20.15 两次硬币

（条件概率）
扔一枚硬币，第一次是正面的概率是多少?
第一次是正面，问第二次是正面的概率是多少?
对以下两种情况分别讨论

1. 一枚正常的硬币。

2. 从两枚硬币中随机选出一枚，其中一枚两面都是正面，一枚两面都是反面。

答案

第一个问题答案都是 0.5。
第二个问题的答案，分别是 0.5 和 1。
因为两次硬币正面并不独立。

20.16 硬币

一个硬币，前 100 次均为正面。
求第 101 次为正面的概率。
看知乎学知识系列
第一种答案 $\frac{1}{2}$，我们认为硬币是正面的概率为 $p = \frac{1}{2}$。
第二种答案

$$\frac{\int_0^1 x^{101} \mathrm{d}x}{\int_0^1 x^{100} \mathrm{d}x} = \frac{101}{102}$$

我们认为硬币是正面的概率 p 是在 0 到 1 之间均匀分布的，然后用积分计算条件概率。

20.17 天选之人

为了简化题目，我们认为每个人的分数是一个 0 到 1 之间随机的实数。
某位同学考试完之后问其他同学的分数。
问了 100 个人，发现自己的分数都更高。
问自己比第 101 个人分数更高的概率?

答案

$$\Pr[\text{比第 101 个人分高}|\text{比前 100 个人分高}] = \Pr[\text{比前 101 个人分高}]/\Pr[\text{比前 100 个人分高}]$$

$$\Pr[\text{比前 101 个人分高}] = \int_0^1 x^{101}\mathrm{d}x = \frac{1}{102}$$

$$\Pr[\text{比前 100 个人分高}] = \int_0^1 x^{100}\mathrm{d}x = \frac{1}{101}$$

所以最终答案是 $\frac{101}{102}$。
这个结论可以推广，如果你问了 $p+q$ 个人，你比其中 p 个人高，比其中 q 个人低。
那么你比下一个人高的概率是 $\frac{p+1}{p+q+2}$。

20.18 摸牌

扑克牌 52 张，其中 26 红，26 黑。
随机洗牌。
你一张一张摸，如果是红色，你获得 1 元，如果是黑色，你失去 1 元。
你可以随时叫停。
问期望收益。

答案

首先答案是不小于 0。
因为无论何种逆境，你都可以选择摸完，最后收益为 0。
然后可以模拟 $2, 4, 6$ 张的情况。
发现并没有规律。
最后决定写一个 DP，就是没有规律。
对于 52 张，答案大约是 2.624475548994。

20.19 计划生育

如果认为所有孩子性别随机且独立。
如果每个家庭都持续生育，直到出现一个男孩。
那么社会上的男女比例会是多少?

答案

当然是 $1:1$。
注意，这个和每个家庭的男女比例平均是多少不一样。

20.20 时代不同了，男女都一样

如果认为每天出生概率相同。（虽然由于政策，季节，并不相同。）
并且认为任意两个人生日相互独立。
并且认为所有孩子性别随机且独立。
并且不考虑闰年。
某人家有两个孩子，已知其中一个是女生，问另外一个是男生的概率?
某人家有两个孩子，已知其中一个星期日出生的女生，问另外一个是男生的概率?
某人家有两个孩子，已知其中一个圣诞节出生的女生，问另外一个是男生的概率?

答案

$$\frac{2}{3}$$

$$\frac{14}{27}$$

$$\frac{730}{1459}$$

因为基本事件并不相同。

20.21 奇妙的高精度

罗雨屏多年前给我讲的一道题
输入 x，计算 $(1+x+x^2+x^4) \bmod 10007$ 的结果。
只允许用加法，减法，取模，（和用变量存下运算结果）。
问最少需要几次运算?

答案

主要是如何乘法。
注意到 $M^2 \bmod (M-x) = x^2$，只要满足 $M-x > x^2$ 即可。
注意到我们可以先计算 $x \bmod 10007$，这样 x 和 x^2 就有范围了。
可以得到一个九次的做法。
注意到 $f(M) \bmod (M-x) = f(x)$，如果 $f(x)$ 是多项式，并且 $M-x > f(x)$。
可以得到一个四次的做法。

21 bzoj 1426

21.1 题目背景

bzoj 1426 收集邮票

21.2 题目大意

有 n 种不同的邮票，皮皮想收集所有种类的邮票。唯一的收集方法是到同学凡凡那里购买，每次只能买一张，并且买到的邮票究竟是 n 种邮票中的哪一种是等概率的，概率均为 $1/n$。但是由于凡凡也很喜欢邮票，所以皮皮购买第 k 张邮票需要支付 k 元钱。现在皮皮手中没有邮票，皮皮想知道自己得到所有种类的邮票需要花费的钱数目的期望。

21.3 题目分析

考虑第 k 张邮票需要 $x+k$ 的钱，计算当前已经抽到 t 个，接下来剩余的期望抽取次数和期望花费是多少。

设 $p = \frac{n}{n-t}$

期望次数：$\frac{1}{p}$。

期望花费：$\sum_{i=1}(x+i)(1-p)^{i-1} = \frac{1+px}{p^2}$

22 bzoj 2554

22.1 题目背景

bzoj 2554 color

22.2 题目大意

22.3 题目分析

23 高斯消元

对于解方程来说，可以分为三种情况。

1. 整数
2. 实数
3. 异或

23.1 实现技巧

对于实数的，为了避免精度误差，一般是把绝对值最大的换到当前行，用当前行去消别的行。

消去别的行的时候，不仅可以消去之后的，也可以消去之前的，这样最后算解的时候，会容易许多。

对于异或版本，可以用 bitset 优化。

23.2 一类问题

poj 1222

枚举第一行的情况，推出之后的情况。

23.3 一些结论

如果系数矩阵可逆，那么一定有唯一解。
矩阵可逆，等价于行列式值不为 0。
如果矩阵不可逆，可能无穷多解，或者无解。
形式一定是特解加通解的形式。
矩阵的秩决定了自由元的个数。

23.4 行列式

高斯消元也可以用来求行列式的值，不过那是另一个故事了。

24 Luogu P3389

24.1 题目背景

P3389 【模板】高斯消元法

24.2 题目大意

给定一个线性方程组，对其求解。
实数。

24.3 题目解法

每个方程形式类似于 $a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$

24.4 参考代码

参考代码

25 Luogu P4783

25.1 题目背景

P4783 【模板】矩阵求逆

25.2 题目大意

求一个 $n \times n$ 的矩阵的逆矩阵。答案对 1000000007 取模。

25.3 题目解法

在后面接一个单位矩阵，然后进行初等行变换。
这个题教育我们，取模的数字必须用 const，否则不会触发优化！

25.4 参考代码

参考代码

26 bzoj 1013

26.1 题目背景

[JSOI2008] 球形空间产生器 sphere

26.2 题目大意

26.3 题目分析

解实数方程。

27 bzoj 4004

27.1 题目背景

bzoj 4004 [JLOI2015] 装备购买

27.2 题目大意

选择价钱尽可能少的向量，使得所有向量都能被线性表示出来。

27.3 参考题解

向量的个数是一定的。
按权值排序，贪心即可。
中间需要高斯消元来判断。

28 spoj JZPLIT

28.1 题目背景

Turn on the lights

28.2 题目大意

28.3 题目解法

29 spoj JZPLIT2

29.1 题目背景

Turn on the lights 2

29.2 题目大意

29.3 题目解法

29.4 网上版本

这个游戏可以在 hacker.org 上玩。

30 行列式

30.1 基本性质

1. 交换两行，行列式的值改变符号。
2. 其中一行加上另一行的若干倍，行列式的值不变。

30.2 面积/体积/高维体积

叉积，混合积。

30.3 Matrix Tree Theorem

数无向图，生成树的个数。
Wikipedia

30.4 BEST Theorem

数有向图，欧拉回路的个数。
Wikipedia

30.5 引入变量

冬令营中提到过这个题目的应用。

31 bzoj 1964

31.1 题目背景

hull 三维凸包

31.2 题目大意

输入 5 个点，求凸包体积。

31.3 题目分析

任取其中 4 个点，算出四面体体积。
五个体积求和，除以二，就是答案。

32 spoj HIGH

32.1 题目背景

Highways

32.2 题目大意

输入 $n \leq 12$ 个点的图。
保证答案在 long long 范围内。

32.3 题目解法

按照定义计算即可，容易越界。
直接 double 消元
辗转相除计算行列式

33 spoj DETER3

33.1 题目背景

Find The Determinant III

33.2 题目大意

行列式求值。

33.3 题目解法

论文题

34 Codeforces 917D

34.1 题目背景

Stranger Trees

34.2 题目大意

输入一个 $n \leq 100$ 个点的树。
问有多少棵树和这个数的交集恰好是 k 个边。
对于每个 $k(0 \leq k \leq n-1)$ 都输出答案。

34.3 题目分析

上分的绝佳场次。就是去年冬令营讲过的题目。
把原有的边看成 x，原没有的边看成 1。
用矩阵树定理求行列式的值，这个值也是一个多项式。
这个多项式的 k 次项，就是交集恰好为 k 的树的个数。
多项式运算比较复杂，采用先选 n 个位置求值，最后插值回多项式的方法。

35 矩阵与线性递推

35.1 简介

你可以认为是一个二维数组。
特别的，在信息学中有用的，几乎只有方阵。

35.2 矩阵乘法

矩阵乘法满足结合律。

35.3 应用

最基础的应用是计算线性递推。

35.4 矩阵求逆

如果矩阵的行列式不为 0 那么可以求逆。
需要高斯消元。

35.5 线性递推

矩阵可以用来求解线性递推，需要同时维护 k 个变量的线性递推，时间复杂度为 $O(k^3 \log t)$。

36 ProjectEuler 258

36.1 题目背景

A lagged Fibonacci sequence

36.2 题目大意

定义数列 g_n，如下：

$$g_i = \begin{cases} 1 & 0 \le i \le 1999 \\ g_{i-2000} + g_{i-1999} & i \ge 2000 \end{cases}$$

求 $n = 10^{18}$ 时的 g_n 结果对 20092010 取模。

36.3 题目分析

$O(k^2 \log n)$，可以在时限内出解。
简而言之，就是在模 20092010 的前提下，计算 $x^{10^{18}} \bmod (x^{2000} - x - 1)$ 的结果，设为

$$\sum_{i=0}^{1999} r_i x^i$$

计算 $\sum_{i=0}^{1999} r_i g_i$ 就是答案，相当于直接将 $x = 1$ 带入求值。
其中多项式取模的部分可以有很多种实现方法，比如暴力倍增，或者是用 Sage 之类的语言。

36.4 Cayley–Hamilton 定理

首先我们陈述 Cayley-Hamilton 定理
考虑转移矩阵 M 的特征多项式 $p(x)$，我们有 $p(M) = 0$。
在本题中，即有 $M^{2000} - M - I = 0$。这个形式非常喵，他表示

$$M^{10^{18}} = f(M)(M^{2000} - M - I) + g(M)$$

其中 $f(M)$ 是一个关于 M 的多项式，相当于商。
$g(M)$ 是一个关于 M 小于 2000 次的多项式，相当于余数。
第一部分一定是 0，我们可以根据第二部分 $g(M)$ 和初始值计算出最终的答案。

36.5 特征多项式

我们设 A 为 $n \times n$ 的方阵，I 是 $n \times n$ 的单位矩阵，那么 A 的特征多项式就是

$$p(\lambda) = \det(\lambda I - A)$$

其中 det 表示行列式求值。
在本题中，特征多项式为 $x^{2000} - x - 1$。

37 ProjectEuler 458

37.1 题目背景

Permutations of Project

37.2 题目大意

字符集大小是 7，问有多少个长度为 n 的字符串，7 个字符的任意排列，都不能作为子串出现。
换句话说，任意连续 7 个字符，均不包含全部 7 个字符。

37.3 题目分析

设字符集大小为 k。

设答案为 t_n，除此之外，定义数列 s_n。

s_n 为除了最后 k 个字符之外，均满足条件，并且最后 k 个字符是一个指定的排列。

显然无论指定哪个排列，答案都是一样的。

有两个关系式

$$kt_{n-1} = t_n + k!s_n$$

即对于任意合法方案，再加一个字符之后，可能形成一个新的合法方案，或者是最后 k 个字符构成一个排列。

$$t_n = \sum_{i=1}^{k}(k-i)!s_{n+i}$$

即对于任意合法方案，依次加入 k 个字符，在第 i 个字符加入之后，最后 k 个字符构成一个排列。

这样根据初始值

$$t_i(0 \le i < k) = k^i$$

$$s_i(0 \le i < k) = 0$$

37.4 生成函数

设答案的生成函数为 T。

设最后 7 位是一个排列，并且除了最后 7 位，均满足题目要求的条件的生成函数为 S。

$$1 + 7xT = T + 7!S$$

$$x^7T = S + 1!xS + 2!x^2S + \cdots + 6!x^6S$$

解得

$$T = \frac{1 + x + 2x^2 + 6x^3 + 24x^4 + 120x^5 + 720x^6}{1 - 6x - 5x^2 - 8x^3 - 18x^4 - 48x^5 - 120x^6}$$

38 hdu 4914

38.1 题目背景

Linear recursive sequence

38.2 题目大意

模板题。

38.3 题目分析

多项式取模。

倍增。

注意到主要有两部分，多项式乘法，多项式取模。

因为取模的多项式只有三项，所以多项式取模可以暴力。

多项式乘法用 FFT。

39 bzoj 1494

39.1 题目背景

[NOI2007] 生成树计数

39.2 题目大意

n 个点构成一条链，如果两个点之间的距离 $\leq k$，那么他们之间有边。

问生成树个数。

39.3 题目解法一

当时的官方解法大概如下。

如果会行列式对质数取模，用高斯消元求值，便可以得到 60 分的高分。

第 7 个点和第 8 个点本意是面相写出 $O(n)$ 状态压缩的选手的，但是实际上行列式写得好也能过。

如果想通过这个题，需要想到状态压缩的做法。

考虑存下最后 k 个点的连通性，对于 $k = 2, 3, 4, 5$ 分别有 $2, 5, 15, 52$ 种可能。

这个是贝尔数，或者说是第二类斯特灵数的求和。

之后转移就枚举下一个点和之前 k 个点的连接情况共 2^k 种可能性。

不能出现环。

如果即将抛弃的点和其他 $k-1$ 个点均不连通，那么必须和新加进来的点连通。

生成转移矩阵，转移即可。

39.4 题目解法二

出题人以为题目中抛出一个看似无用的行列式，可以转移大家焦点，隐藏题目真实解法。但是实际上这个题在近年有了新的做法，行列式的计算复杂度是 $O(n^3)$。

通过行列式可以得到前几百项的结果。

设想这个答案可以通过矩阵乘法求出，那么他必然是一个 ≤ 52 阶的递推数列。

不妨设递推数列为

$$f_n = \sum_{1 \leq i \leq d} a_i f_{n-i}$$

你可以直接用前 $2n+1$ 项，列出 n 个方程，解出答案，但是略微麻烦。

解方程，除了解出唯一解，还有两个可能，无穷多解和无解。

40 多项式

给定一个环 R （R 通常是交换环，可以是有理数、实数或者复数等等）以及一个未知数 X，则任何形同：

$$a_0 + a_1X + \cdots + a_{n-1}X^{n-1} + a_nX^n$$

的代数表达式叫做 R 上的一元多项式。其中 $a_0, a_1, \cdots, a_n$ 是 R 中的元素。未知数不代表任何值，但环 R 上的所有运算都对它适用。

对于一元多项式来说，未知数在多项式各项中最大的次数称为多项式的次数。

注意 n 次的多项式最多可以有 $n+1$ 项。

40.1 多项式加法减法

两个多项式相加可以看作是对两组单项式的和进行重组与合并同类项。通过加法结合律，可以将同类项放在一起，合并之后就得到了两个多项式的和。

40.2 多项式乘法

计算两个多项式相乘时，首先使用乘法对加法的分配律将各项拆出，然后运用乘法结合律整合每一项，最后和加法一样整合同类项，就能得到乘积多项式。

40.3 多项式求导

按照定义做即可 $(x^n)' = nx^{n-1}$。

40.4 多项式积分

按照定义做即可 $\int x^n \mathrm{d}x = \frac{n+1}{x}^{n+1}$。

一般认为积分后不加常数项（常数项为 0）

40.5 多项式除法取模

和整数之间的带余除法类似，一元多项式之间也可以进行带余除法。可以证明，设有多项式 $A(x)$ 和非零多项式 $B(x)$ ，则存在唯一的多项式 $Q(x)$ 和 $R(x)$，满足：

$$A(x) = Q(x)B(x) + R(x)$$

也可以简单地认为

$$A(x) \bmod B(x) = R(x)$$

40.6 多项式求逆

显然对于次数大于 1 的多项式 $A(x)$，不可能找到多项式 $B(x)$ 使得

$$A(x)B(x) = 1$$

和数论的情况类似，如果

$$A(x)B(x) = 1 \pmod{x^n}$$

那么称 $B(x)$ 为 $A(x)$ 在 $\mathrm{mod} x^n$ 意义下的逆元。

40.7 多项式开根

和数论的情况类似，如果

$$B^2(x) = A(x) \pmod{x^n}$$

那么称 $B(x)$ 为 $A(x)$ 在 $\mathrm{mod} x^n$ 意义下的平方根。

40.8 信息学竞赛相关

为了计算精确，题目一般对质数取模。

为了可以使用 FFT 或 NTT，n 一般是 2 的次幂，质数减一之后是 n 的倍数。即存在 n 次单位根。

如果 n 不是 2 的次幂，可以算出一个比 n 大的 2 的次幂结果，只取前 n 项。

如果取模的数字性质不好，需要比较麻烦的三模数 NTT，或者拆系数 FFT。

40.9 多项式求逆

假设 n 是 2 的次幂，可以递归解决这个问题。

如果 n 不是 2 的次幂，可以算出一个比 n 大的 2 的次幂结果，只取前 n 项。

如果 $n = 1$，那么 $A(x)$ 只有常数项，如果非零，直接乘法逆元，如果为零，那么无解。

如果 $n > 1$，先求出 $n/2$ 的答案，即

$$A(x)B_0(x) = 1 \pmod{x^{n/2}}$$

注意到对于要求的 $B(x)$ 满足

$$A(x)B(x) = 1 \pmod{x^n}$$

也满足

$$A(x)B(x) = 1 \pmod{x^{n/2}}$$

相减可以得到

$$A(x)(B(x) - B_0(x)) = 0 \pmod{x^{n/2}}$$

因为 $A(x)$ 可逆，所以可以消去

$$(B(x) - B_0(x)) = 0 \pmod{x^{n/2}}$$

两侧平方，取模的数字同样平方

$$B(x)^2 - 2B(x)B_0(x) + B_0^2(x) = 0 \pmod{x^n}$$

两遍同时乘以 $A(x)$，注意到 $B(x)$ 是 $A(x)$ 的逆元，他们乘积为 1。

$$B(x) - 2B_0(x) + A(x)B_0^2(x) = 0 \pmod{x^n}$$

移项之后得到

$$B(x) = 2B_0(x) - A(x)B_0^2(x) \pmod{x^n}$$

总复杂度为 $T(n) = T(n/2) + O(\text{多项式乘法})$。

如果使用 $O(n \log n)$ 的多项式乘法，那么复杂度为 $O(n \log n)$。

如果使用 $O(n^2)$ 的多项式乘法，那么复杂度为 $O(n^2)$。

40.10 多项式开根

假设 n 是 2 的次幂，可以递归解决这个问题。

如果 n 不是 2 的次幂，可以算出一个比 n 大的 2 的次幂结果，只取前 n 项。

如果 $n = 1$，那么 $A(x)$ 只有常数项，如果二次剩余，直接二次剩余，如果二次非剩余，直接二次非剩余。

如果 $n > 1$，先求出 $n/2$ 的答案，即

$$B_0^2(x) = A(x) \pmod{x^{n/2}}$$

注意到对于要求的 $B(x)$ 满足

$$B^2(x) = A(x) \pmod{x^n}$$

也满足

$$B^2(x) = A(x) \pmod{x^{n/2}}$$

相减可以得到

$$(B(x) + B_0(x))(B(x) - B_0(x)) = 0 \pmod{x^{n/2}}$$

像整数开根一样，会有两个解，不妨认为后者为零

$$(B(x) - B_0(x)) = 0 \pmod{x^{n/2}}$$

两侧平方，取模的数字同样平方

$$B(x)^2 - 2B(x)B_0(x) + B_0^2(x) = 0 \pmod{x^n}$$

将 $B(x)^2$ 替换为 $A(x)$

$$A(x) - 2B(x)B_0(x) + B_0^2(x) = 0 \pmod{x^n}$$

移项之后得到

$$B(x) = \frac{A(x) + B_0^2(x)}{2B_0(x)} \pmod{x^n}$$

除法部分直接乘以分母的逆元。

总复杂度为 $T(n) = T(n/2) + O(\text{多项式乘法}) + O(\text{多项式求逆})$。

如果使用 $O(n \log n)$ 的多项式乘法，那么复杂度为 $O(n \log n)$。

如果使用 $O(n^2)$ 的多项式乘法，那么复杂度为 $O(n^2)$。

40.11 多项式除法取模

对于一个 n 次多项式 $A(x)$，设 $A^R(x) = x^n A(1/x)$ 即把 x 换为 $\frac{1}{x}$，再乘以 x^n；这一步相当于把多项式的系数前后 `reverse`。

设 $A(x)$ 是 n 次多项式，$B(x)$ 是 m 次多项式，那么 $Q(x)$ 是 $n-m$ 次多项式，$R(x)$ 的次数小于 m。

$$A(x) = Q(x)B(x) + R(x)$$

把 x 换为 $\frac{1}{x}$，再乘以 x^n。

$$x^n A(\frac{1}{x}) = x^{n-m} Q(\frac{1}{x}) x^m B(\frac{1}{x}) + x^{n-m+1} x^{m-1} R(\frac{1}{x})$$

替换成 A^R，可以得到

$$A^R(x) = Q^R(x)B^R(x) + x^{n-m+1}R^R(x)$$

两边同时对 x^{n-m+1} 取模，可以得到

$$A^R(x) = Q^R(x)B^R(x) \pmod{x^{n-m+1}}$$

而 Q^R 恰好是一个 $n-m$ 次的多项式，注意到 $B^R(x)$ 非零，可以求逆元。

$$Q^R(x) = \frac{A^R(x)}{B^R(x)} \pmod{x^{n-m+1}}$$

求出 $Q(x)$ 后，可以直接计算得到 $R(x) = A(x) - Q(x)B(x)$。

总复杂度为 $O(\text{多项式乘法}) + O(\text{多项式求逆})$。

如果使用 $O(n \log n)$ 的多项式乘法，那么复杂度为 $O(n \log n)$。

40.12 多项式 Log

$$\log(1+x) = \sum_{n=1}^{+\infty} \frac{(-1)^{n+1}}{n} x^n$$

对于多项式 $A(x)$ 来说

$$\log(A(x)) = \log(1 + (A(X) - 1)) = \sum_{n=1}^{+\infty} \frac{(-1)^{n+1}}{n} (A(x) - 1)^n$$

特别注意，如果 $A(x) - 1$ 常数项非 0，是不能取对数的。

$$(\log(A(x)))' = \frac{A'(x)}{A(x)}$$

$$\log(A(x)) = \int \frac{A'(x)}{A(x)} \mathrm{d}x$$

用定义暴力算的复杂度为 $O(n^2)$。

总复杂度为 $O(\text{多项式乘法}) + O(\text{多项式求逆})$。

如果使用 $O(n \log n)$ 的多项式乘法，那么复杂度为 $O(n \log n)$。

40.13 多项式 Exp

$$\exp(x) = \sum_{n=1}^{+\infty} \frac{x^n}{n!}$$

对于多项式 $A(x)$ 来说

$$\exp(A(x)) = \sum_{n=1}^{+\infty} \frac{A^n(x)}{n!}$$

特别注意，如果 $A(x)$ 常数项非 0，是不能取指数的。

对于一个已知的函数 $G(F(x))$，求多项式 $F(x)$ 使得 $G(F(x)) = 0$。

注意 $G(F(x))$ 是以多项式 $F(x)$ 为参数的函数，而不是函数的嵌套。

用类似于倍增的牛顿迭代，首先求出 $F(x)$，使得

$$G(F(x)) = 0 \pmod x$$

如果已知 $G(F_0(x)) = 0 \pmod{x^n}$，可以将 G 在 $F_0(x)$ 的位置展开。

$$G(F(x)) = G(F_0(x)) + G'(F_0(x)) \times (F(x) - F_0(x)) + \frac{G''(F_0(x)) \times (F(x) - F_0(x))^2}{2!} + \cdots$$

其中 $G(F(x)) = 0 \pmod{x^{2n}}$，显然 $(F(x) - F_0(x)) = 0 \pmod{x^n}$。（显然吗?）

那么 $(F(x) - F_0(x))^2 = 0 \pmod{x^{2n}}$，所以二次项及之后的部分都不需要考虑。

$$G(F_0(x)) + G'(F_0(x)) \times (F(x) - F_0(x)) = 0 \pmod{x^{2n}}$$

解得

$$F(x) = F_0(x) - \frac{G(F_0(x))}{G'(F_0(x))} \pmod{x^{2n}}$$

在本题中 $G(F(x)) = \log(F(x)) - A(x), G'(F(x)) = \frac{1}{F(x)}$，所以

$$F(x) = F_0(x) - F_0(x)(\log(F_0(x)) - A(x)) \pmod{x^{2n}}$$

$$F(x) = F_0(x)(1 - \log(F_0(x)) + A(x)) \pmod{x^{2n}}$$

$F_0(x)$ 只要次数较低的 n 项正确即可，前面是否正确无所谓。

用定义暴力算的复杂度为 $O(n^2)$。

总复杂度为 $T(n) = T(n/2) + O(\text{多项式乘法}) + O(\text{多项式求逆})$。

如果使用 $O(n \log n)$ 的多项式乘法，那么复杂度为 $O(n \log n)$。

40.14 多项式开 k 次方

如果多项式的常数项为 1，那么可以通过取 Log，除以 k，再 Exp 的方法来做。
如果常数项不为 1，那么需要提出常数项，单独对常数项做一次开 k 次方。

40.15 泰勒展开

泰勒展开
建议记住常见的泰勒展开。

40.16 拉格朗日插值

拉格朗日插值法
OI Wiki
关键点都差不多，主要有两类问题。已知 n 个点 (x_i, y_i)，确定一个 $n-1$ 次多项式。

- 求出这个多项式在一个新的点的值。
- 求出多项式每一项的系数。

拉格朗日插值在特殊情况下，可以推广到多个变量的情况，但是几乎没有用。
不妨考虑多种颜色的边的生成树计数。

40.17 多项式多点求值

输入一个 $n-1$ 次多项式 $A(x)$ 的系数，和一个 x_0，求 $A(x_0)$，可以用秦九韶算法。
如果对多个点 $x_0, x_1, \ldots, x_{n-1}$ 同时求值呢？暴力是 $O(n^2)$ 的。
把要求值的点任意分成两部分，比如

$$X_0 = \{x_0, x_1, \ldots, x_{n/2-1}\}, X_1 = \{x_{n/2}, x_{n/2+1}, \ldots, x_{n-1}\}$$

定义多项式

$$P_0(x) = \prod_{x_i \in X_0} (x - x_i), P_1(x) = \prod_{x_i \in X_1} (x - x_i)$$

计算 $P_0(x)$ 和 $P_1(x)$ 可以递归和分治 FFT 来计算。
对于 X_i 里的数，只需要带入 $A(x) \bmod P_i(x)$ 即可。
多项式取模时间复杂度 $O(n \log n)$。
即 $T(n) = 2T(n/2) + O(n \log n) = O(n \log^2 n)$。

40.18 多项式快速插值一

快速插值 $O(n \log^2 n)$

40.19 多项式快速插值二

快速插值 $O(n\log^3 n)$
输入 n 个点 (x_i, y_i)，希望求一个 $n-1$ 次多项式，通过所有点。
把 n 个点任意分成两部分，比如 $X_0 = \{x_0, x_1, \ldots, x_{n/2-1}\}$ 和 $X_1 = \{x_{n/2}, x_{n/2+1}, \ldots, x_{n-1}\}$。
对于 X_0 求出对应的多项式 $A_0(x)$。
定义多项式 $P_0(x) = \prod_{x_i \in X_0}(x - x_i)$。
最终答案一定是

$$A(x) = B_1(x)P_0(x) + A_0(x)$$

关键在于 $B_1(x)$ 是什么。
也就是说，对于 $x_i \in X_1$，有 $y_i = B_1(x_i)P_0(x_i) + A_0(x_i)$。
可以用快速求值，求出 $P_0(x_i)$ 和 $A_0(x_i)$。
可得 $B_1(x_i) = \frac{y_i - A_0(x_i)}{P_0(x_i)}$。
递归求解 $B_1(x)$ 即可。
多点求值的复杂度为 $O(n\log^2 n)$。
即 $T(n) = 2T(n/2) + O(n\log^2 n) = O(n\log^3 n)$。

41 范浩强的好题

41.1 题目背景

范浩强大神教我的，不过这个题因为种种原因最后没有被出出来。

41.2 题目大意

考虑一个数组 $\{x_i\}$。
支持两个操作。

- 区间加法
- 每个 x_i 可以对应到单位圆上一个正实数 y_i，求所有 y_i 的和。

保证任意时刻 $|x_i| \le \frac{1}{2}$。
数组长度 n 和操作数 m 大约 50000。
精度要求 10^{-6}。

41.3 题目解法

注意到所求的 $y_i = \sqrt{1 - x_i^2}$。

$$\sqrt{1 - x_i^2} = 1 - \frac{1}{2}x_i^2 - \frac{1}{8}x_i^4 - \frac{1}{16}x_i^6 - \cdots$$

所以分别维护区间的 0 次方到 10 次方即可。

41.4 其他

不过据说清华冬令营出过一次 e^x 展开的题目。
这类题目的关键在于 x 必须非常小，否则收敛非常慢。

42 罗雨屏的好题

42.1 题目背景

UyHiP-2013-Jun

42.2 题目大意

想说的罗雨屏都说了。

43 Luogu P2312

43.1 题目背景

P2312 解方程

43.2 题目大意

相信大家都做过。

43.3 题目解法

相信大家都做过。
数据没有卡取模的数字。
这个坑也讲过。（Freda 城堡的密码）

44 bzoj 2742

44.1 题目背景

bzoj 2742 [HEOI2012]Akai 的数学作业

44.2 题目大意

输入一个整系数多项式方程，求所有有理数解，按分数输出，重根只输出一次。
整系数绝对值范围 20000000，方程次数 n，不取模。

44.3 题目分析

首先如果 $a_0=0$，整个方程可以除以 x。
注意到有理数解的分子一定是 a_0 的约数，分母一定是 a_n 的约数。
枚举两个约数，枚举正负，带入验证即可。
越界的问题和 NOIP2014 解方程处理方法一样，对大数取模即可。
数据只卡了自然溢出。

45 Codeforces 622F

45.1 题目背景

The Sum of the k-th Powers

45.2 题目大意

输入 n,k，求 1 到 n 的 k 次方和。

45.3 题目分析

大概两种做法。
差分或斯特林数，时间复杂度 $O(k^2)$。
拉格朗日插值，时间复杂度 $O(k)$ 或者 $O(k\log k)$。

45.4 差分/斯特林数

举例来说，考虑数列
$1,8,27,64,125,216,\ldots$
求差分可得
$7,19,37,61,91,\ldots$
$12,18,24,30,\ldots$
$6,6,6,\ldots$
$0,0,\ldots$
每个数列的第一项为 $1,7,12,6$。所以最终的答案即为

$$\binom{n}{1}+7\binom{n}{2}+12\binom{n}{3}+6\binom{n}{4}=\frac{n^2(n+1)^2}{4}$$

另一种解释，这个相当于利用了次方和组合数的关系

$$x^k=\sum_i S(k,i)i!\binom{x}{i}$$

这个题在 JZPTREE 中也有应用。

$$1=S(4,1)\times 0!, 7=S(4,2)\times 1!, 12=S(4,3)\times 2!, 6=S(4,4)\times 3!$$

45.5 拉格朗日插值法

首先根据一些常识我们可以知道答案是一个 $d = k + 1$ 次的多项式，需要先通过暴力求出这个多项式在 $0, 1, \ldots, d$ 这 $d + 1$ 个点的值，再计算这个多项式在 n 这个点的值。

$$f(n) = \sum_{i=0}^{d} (-1)^{d-i} f(i) \frac{n(n-1)\cdots(n-d)}{(d-i)!i!(n-i)}$$

注意到 x^k 是一个积性函数，也就是说计算 1^k 到 k^k 只需要 $O(k)$ 的时间，即 $O(k/\log k)$ 个质数 x 的 $f(x)$，计算每个质数需要 $O(\log k)$ 的时间，其他的数通过积性函数计算。

然后计算上面的表达式只需要结合预处理即可以做到 $O(k)$。

对于上面的表达式 $n(n-1)\cdots(n-d)$ 没有 $(n-i)$。

如果 n 性质比较好，可以用逆元实现，如果性质不好，只能用前缀乘积和后缀乘积。

46 hdu 4651

46.1 题目背景

Partition

46.2 题目大意

把 n 拆分成若干个自然数的和，问有多少种方案。

拆出的自然数不计顺序，比如 $3 = 1 + 2 = 2 + 1$ 算作同一种方案。

$(n \leq 100000)$

46.3 题目分析

OEIS A000041 the partition numbers

一个显而易见的做法是无穷背包。

设 $P_x(n)$ 为 n 的拆分数。

拆分数的生成函数为

$$P(x) = P_x(0)x^0 + P_x(1)x^1 + P_x(2)x^2 + \cdots$$

$$P(x) = (1 + x + x^2 + \cdots)(1 + x^2 + x^4 + \cdots)(1 + x^3 + x^6 + \cdots)\cdots$$

$$P(x) = \prod_{i=1}^{+\infty}(1 + x^i + x^{2i} + \cdots) = \prod_{i=1}^{+\infty} \frac{1}{1 - x^i}$$

欧拉函数（复变函数）

$$\phi(x) = \prod_{k=1}^{+\infty}(1 - x^k)$$

五边形数定理

$$\phi(x)=\prod_{k=1}^{+\infty}(1-x^k)=\sum_{n=-\infty}^{+\infty}(-1)^n x^{(3n^2-n)/2}$$

$$\phi(x)=1-x-x^2+x^5+x^7-x^{12}-x^{15}+x^{22}+x^{26}-\cdots$$

欧拉函数和拆分数的生成函数之间有非常简单的关系

$$\phi(x)P(x)=1$$

显然 $P_x(0)=1$。对于 $n>0$，有

$$P_x(n)-P_x(n-1)-P_x(n-2)+P_x(n-5)+P_x(n-7)-\cdots=0$$

省略号部分，直到 P_x 的参数为负数为止。

左边有 $O(\sqrt{n})$ 项，可以通过小于 n 的项计算出 $P_x(n)$。

47 hdu 4658

47.1 题目背景

Integer Partition

47.2 题目大意

把 n 拆分成若干个自然数的和，相同的数字，不能出现 k 次或更多，问有多少种方案。

拆出的自然数不计顺序，比如 $3=1+2=2+1$ 算作同一种方案。

$(1\le n,k,T\le 100000)$

T 组数据。

47.3 题目分析

注意是相同的数字不能出现 k 次或更多，而不是拆分成至多 $k-1$ 个数字。

和刚才类似，考虑生成函数。

答案的生成函数为

$$F(x)=F_x(0)x^0+F_x(1)x^1+F_x(2)x^2+\cdots$$

$$F(x)=(1+x+x^2+\cdots+x^{k-1})(1+x^2+x^4+\cdots+x^{2(k-1)})(1+x^3+x^6+\cdots+x^{3(k-1)})\cdots$$

$$F(x)=\prod_{i=1}^{+\infty}(1+x^i+x^{2i}+\cdots+x^{i(k-1)})=\prod_{i=1}^{+\infty}\frac{1-x^k}{1-x^i}$$

欧拉函数和答案的生成函数之间有非常简单的关系

$$F(x)=\frac{\phi(x^k)}{\phi(x)}=\phi(x^k)P(x)$$

其中 $P(x)$ 是拆分数。那么 $F(n)=P_x(n)-P_x(n-k)-P_x(n-2k)+P_x(n-5k)+\cdots$。

这样得到了一个 $O(n\sqrt{n})$ 预处理，单次回答 $O(\sqrt{n})$ 的做法。

48 hdu 3205

48.1 题目背景

Factorization

48.2 题目大意

对 x^n-1 分解因式。

48.3 题目分析

分圆多项式

$\Phi_n(x)$ 是 x^n-1 的一个因式，并且不是任何 $x^k-1(k<n)$ 的因式。

$$\Phi_n(x)=\prod_{1\le k\le n,\gcd(k,n)=1}(x-e^{\frac{k}{n}2i\pi})$$

$$x^n-1=\prod_{d|n}\Phi_d(x)$$

所以从小到大依次计算 n 的所有约数的分圆多项式即可。

49 hdu 6414

49.1 题目背景

带劲的多项式

49.2 题目大意

解方程

$$f(x)=(x-\lambda_1)^{l_1}(x-\lambda_2)^{l_2}\cdots(x-\lambda_m)^{l_m}=0$$

解出所有的 λ_i 和 l_i。

保证所有 l_i 互不相同。

输入 $f(x)=a_0x^0+a_1x^1+\cdots+a_nx^n$ 的所有系数 a_n。

所有运算都是在模 998244353 下进行的。

49.3 题目分析

考虑 $f(x)$ 的导数。

$$f'(x)=\sum_{i=1}^{m}l_i(x_i-\lambda_i)^{l_i-1}\prod_{j\neq i}(x-\lambda_j)^{l_j}$$

考虑 $\gcd(f(x), f'(x))$。

$$\gcd(f(x), f'(x)) = \prod_{i=1}^{m}(x-\lambda_i)^{l_i-1}$$

相当于把每个根的次数都减少 1，当然已经减少到 0 的就不减少了。

50 Codechef BIKE

50.1 题目背景

BIKE

我自己出的，现在感觉这题有点垃圾啊。

50.2 题目大意

大概就是每条边有两个权值 f_i, r_i。

问有多少条路径，起点终点相同，走了 t 条边，走过的所有 f_i 的和模 n 是 a，走过的所有 r_i 的和模 $n-1$ 是 b。对于每个 (a, b) 都输出方案数模 1163962801。

$2 \le n \le 22$

50.3 题目解法

大概就是路径数大家都会算，直接邻接矩阵 t 次方。

然后现在有了这个权值，那么就要引入未知数 x。

有两个权值，就引入两个位置数 x, y。

这个 p 非常特殊，$p-1$ 是 1 到 22 所有数字的倍数。

所以求出原根后，存在需要的所有单位根。

50.4 其他

事实上，几乎所有需要设两个变量的题目，都可以通过一些方式，只设一个变量，这个题也不例外。

51 Codeforces 917D

51.1 题目背景

Stranger Trees

51.2 题目大意

输入一个 $n \le 100$ 个点的树。

问有多少棵树和这个数的交集恰好是 k 个边。

对于每个 $k(0 \le k \le n-1)$ 都输出答案。

51.3 题目分析

上分的绝佳场次。就是去年冬令营讲过的题目。

把原有的边看成 x，原没有的边看成 1。

用矩阵树定理求行列式的值，这个值也是一个多项式。

这个多项式的 k 次项，就是交集恰好为 k 的树的个数。

多项式运算比较复杂，采用先选 n 个位置求值，最后插值回多项式的方法。

52 Codeforces 102028J

52.1 题目背景

CF 102028J. Carpets Removal

52.2 题目大意

在 $m \times m(1 \leq m \leq 1500)$ 的网格上，有 $n(3 \leq n \leq 300000)$ 个地毯。

你可以删掉其中两个地毯，问删除两个之后，未被地毯覆盖的面积最大是多少。

52.3 题目分析

求出每个位置，被几个地毯覆盖了，这个非常简单。

如果某位置只被一个地毯覆盖了，问是谁覆盖的他，这个非常简单，维护地毯编号之和即可。

如果某位置只被二个地毯覆盖了，问是哪两个覆盖的他，这个非常简单，维护地毯编号之和和平方和即可。

53 Fibonacci number

Fibonacci number

定义：

$$F_0 = 0, F_1 = 1$$

$$F_i = F_{i-1} + F_{i-2}(i \geq 2)$$

53.1 求第 n 项

直接矩阵乘法；POJ 3070

或折半递归。

53.2 取模有循环节

根据鸽笼原理，在对 m 取模的前提下，循环节长度至多 m^2，设模 m 最小正循环节为 $p(m)$。

根据 OEIS 的页面，有如下性质。

- 如果 x 和 y 互质，那么 $p(xy) = p(x)p(y)$。

- 对于质数 x，有 $p(x^c) = p(x)x^{c-1}$。
- 对于质数 x，除了 $p(5) = 20$ 之外，其他均满足 $p(x)|(x^2-1)$。

53.3 相关的题目

spoj FRSKH 快速求 Fibonacci 数对任意数取模的循环节。

codechef FN 根据 F_n 直接求出 n，模特定数。

注意到 5 是二次剩余，我们可以找到一个数 x 使得 $x^2 \bmod p = 5$，然后原题变成一个二次方程和离散对数的问题。

CSAcademy fibonacci-mod 根据 F_n, F_{n-1} 求出 n，模任意数。

直接对矩阵进行 Baby Step Gaint Step。

53.4 参考资料

OEIS Fibonacci Number

Wikipedia Fibonacci Number

54 poj 3070

54.1 题目背景

poj 3070 Fibonacci

54.2 题目大意

输出第 n 项 Fibonacci 数模 10000 的结果。

54.3 题目分析

循环节是 15000。

读入 n 直接 ‘n %= 15000‘然后暴力即可。

55 spoj POWERPHI

55.1 题目背景

Power of Phi(medium)

55.2 题目大意

输入 n，输出 $\left(\frac{1+\sqrt{5}}{2}\right)^n$ 最接近的整数。

55.3 题目解法

注意到 $\left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)^n$ 是一个整数数列。
并且第二部分近似为 1。

56 spoj FRS2

56.1 题目背景

FRS2 - Fibonaccibonacci (easy)

56.2 题目大意

输入 n。
输出 $f[f[n]] \bmod 1000000007$。

56.3 题目解法

输出 $f[f[n] \bmod 2000000016] \bmod 1000000007$。

57 spoj FRSKT

57.1 题目背景

FRSKT - Fibonacci recursive sequences (medium)

57.2 题目大意

定义 $G(0, N) = N$，$G(K, N) = G(K-1, F(N))$。
输入 K, N, M 求 $F(K, N) \bmod M$。

57.3 题目解法

每次求循环节即可。

58 Luogu P4000

58.1 题目背景

P4000 斐波那契数列

58.2 题目大意

输出第 n 项 Fibonacci 数模 p 的结果。
$n \le 10^{30000000}, p < 2^{31}$

58.3 题目解法

求循环节，然后直接对 n 取模
循环节性质可以参考题解。

59 FFT

59.1 多项式

多项式乘法 $A(x) = \sum a_i x^i$ 和 $B(x) = \sum b_i x^i$，求 $A(x)B(x)$。

- 系数表示法 直接记录每个系数，
- 点值表示法 记录在 $n+1$ 个不同的位置这个多项式的值。

可以证明，一个点值表示法可以唯一确定一个系数表示法。
两种方法对比

- 多项式求值
$$\Theta(n) < \Theta(?)$$
- 多项式加法
$$\Theta(n) = \Theta(n)$$
- 多项式乘法
$$\Theta(n^2) > \Theta(n)$$
- 多项式整数次方
$$\Theta(kn^2) \gg \Theta(n \lg k)$$

点值表示法是多项式计算的最终归宿。
两种表示方法相互转换。

- 系数表示法到点值表示法。
任取 $n+1$ 个数求值，复杂度 $\Theta(n^2)$。
- 点值表示法到系数表示法。
直接解方程，复杂度 $\Theta(n^3)$
拉格朗日插值法，复杂度 $\Theta(n^2)$

复杂度均不是很理想，但是注意到点值表示法取值的任意性，可以通过合适的取值来优化这个算法。

59.2 复数

约定 $i^2 = -1$。
复平面，一个复数表示平面上的一个点。

两种表示方法，模长幅角，实部虚部，四则运算，相互转换。

欧拉公式

$$e^{i\theta} = \cos\theta + i\sin\theta$$

单位根:

$$\omega_n = \cos\frac{2\pi}{n} + i\sin\frac{2\pi}{n} = e^{2\pi i/n}$$

程序最好实现成一个类。

约定从系数到点值叫做 DFT(Discrete Fourier Transform)。

约定从点值到系数叫做 IDFT(Inverse Discrete Fourier Transform)。

考虑在 $1, \omega_n, \omega_n^2, \omega_n^3, \ldots, \omega_n^{n-1}$ 这些位置的值。

当 $n = 2^k$ 时，在这 n 个点的取值可以快速求出。

考虑计算过程

$$f(x) = \sum_{i=0}^{n-1} a_i(x)^i$$

按奇数和偶数分开

$$f(x) = \sum_{i=0}^{n/2-1} a_{2i}(x)^{2i} + x\sum_{i=0}^{n/2-1} a_{2i+1}(x)^{2i}$$

如果 x 是 n 次单位复根，那么 x^2 一定是 $n/2$ 次单位复根。

所以只需要计算多项式

$$f_0(x) = \sum_{i=0}^{n/2-1} a_{2i}x^i$$

和

$$f_1(x) = \sum_{i=0}^{n/2-1} a_{2i+1}x^i$$

的 DFT，即在所有 $n/2$ 次单位复根的取值。

IDFT 和 DFT 只差一个符号和一个常数。

$$y_k = \sum_{i=0}^{n-1} a_i(\omega^k)^i$$

$$a_k = \frac{1}{n}\sum_{i=0}^{n-1} y_i(\omega^{-k})^i$$

通过先 DFT，乘法，再 IDFT 到多项式系数的方法。

最基本的应用，快速计算卷积

```
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        c[i + j] += a[i] * b[j];
    }
}
```

循环卷积，假设 a 数组和 b 数组长度均为 n。

```
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        c[(i + j) % n] += a[i] * b[j];
    }
}
```

就是循环卷积

59.3 二维 FFT

其实本质上，说是二维循环卷积更为恰当，如果不是循环卷积，可以转化为一维卷积。
一维卷积可以用 FFT 优化，二维也可以优化，但是效果就不明显了。

60 hdu 1402

60.1 题目背景

A * B Problem Plus

60.2 题目大意

高精度 A * B

60.3 题目分析

用 FFT 可以加速计算高精度乘法。
FFT 大概是不进位的高精度乘法，最后手动模拟进位即可。
但是根据经验，如果使用 FFT，那么不能压很多位。(压 2 位已经很多了。)
而如果暴力乘法，轻轻松松压 9 位，所以通过高精度来考查 FFT 不是一个好主意。

60.4 分治乘法

简单介绍分治乘法。
当需要计算 $(A \times 10^{n/2} + B)(C \times 10^{n/2} + D)$ 时。
可以计算 $AC, BD, (A + B) \times (C + D)$。
中间项目 $AD + BC = (A + B)(C + D) - AC - BD$。
时间复杂度 $T(n) = 3T(n/2) + O(n) = n^{\log_2 3} \approx n^{1.59}$。

61 bzoj 2179

61.1 题目背景

FFT 快速傅立叶

61.2 题目大意

高精度乘法。

61.3 题目分析

62 51nod 1028

62.1 题目背景

51nod 1028 大数乘法 V2

62.2 题目大意

高精度乘法。

62.3 题目分析

对于这个题目使用 Go, Python, Ruby, Java 的内置函数都是可以通过的。

63 51nod 1029

63.1 题目背景

51nod 1029 大数除法

63.2 题目大意

64 51nod 1030

64.1 题目背景

51nod 1030 大数进制转换

64.2 题目大意

高精度进制转换，输入一个 36 进制的数字，转化为 10 进制输出。

64.3 题目分析

对于这个题目使用 Go, Python, Ruby, Java 的内置函数都是可以通过的。
如果使用 C++，也可以用通过压位的方式通过。
先把输入的数字转化为 $36^5 = 60466176$ 进制。
然后通过模一次，除一次的方法，转成 $10^{10} = 10000000000$ 进制。
这样可以大大优化常数。

如果使用 FFT 的话，需要分治，即一共 $2n$ 位，将低 n 位和高 n 位分别进行递归转化。然后把高 n 位的结果，乘以 36^n 这是一个高精度乘法，可以用 FFT 优化。

最终总复杂度 $T(n) = 2T(n/2) + O($多项式乘法$)$。

如果使用 $O(n \log n)$ 的多项式乘法，那么复杂度为 $O(n \log^2 n)$。

65 spoj SQRT2

65.1 题目背景

Digits of SQRT(2)

65.2 题目大意

尽可能多的计算 $\sqrt{2}$ 的小数部分。

65.3 题目解法

一个显而易见的想法是二分，基本没用。

另一个想法是牛顿迭代，大概是选择合适的开始的值 x_0，依次计算

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

在这个题中，递推非常特殊

$$x_{n+1} = \frac{1}{2}(x_n + \frac{2}{x_n})$$

牛顿迭代法可以每次获得一倍的精度。

66 spoj PIVAL

66.1 题目背景

Digits of Pi

66.2 题目大意

尽可能多的计算 π 的小数部分。

66.3 题目解法

π 的公式

其中值得引起注意的是

$$\arctan x = \sum_{n=0}^{+\infty} \frac{(-1)^n}{2n+1} x^{2n+1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots$$

$$\frac{\pi}{4} = \arctan\frac{1}{2} + \arctan\frac{1}{3}$$

$$\frac{\pi}{4} = 4\arctan\frac{1}{5} - \arctan\frac{1}{239}$$

利用这 2 个公式，可以非常简单快速的计算 π 的结果。

67 bzoj 2194

67.1 题目背景

bzoj 2194 快速傅立叶之二

bzoj 2194 题面

67.2 题目大意

输入两个长度为 n 的数组 $\{a_i\},\{b_i\}$，下标均从 0 开始。

构造新数组 $c_k(0 \le k < n)$ 为

$$c_k = \sum_{k\le i\le n} a_i b_{i-k}$$

67.3 题目分析

将输入的 b 数组直接 reverse，那么所求的 c_k 为

$$c_k = \sum_{k\le i\le n} a_i b_{n-1-i+k}$$

也就是 a_i 和 b_i 卷积，之后的第 $n-1$ 到 $2n-2$ 共 n 项为答案。

68 bzoj 3160

68.1 题目背景

bzoj 3160

一个非常好的 FFT 和 manache 练习题。

68.2 题目大意

输入一个只包含 A 和 B 的字符串，问有多少个子序列，值和位置都是回文的。

子序列不能连续。（最后把连续的减掉）

68.3 参考题解

把 A 的位置拿出来，自己 FFT 一下。

把 B 的位置拿出来，自己 FFT 一下。

枚举中心，对于每对字母，可以取或者不去，当然不能取空集。

最后减去连续的即可。

menci 的题解

69 bzoj 4503

69.1 题目背景

bzoj 4503 两个串

69.2 题目大意

输入长度为 n 的字符串 S，S 没有通配符。
输入长度为 m 的字符串 T，T 有通配符。
问 T 在 S 的哪些位置出现了。

69.3 参考题解

将 S 转化为数组 f，f_i 为 S_i 的 ASCII 码。
将 T 转化为数组 g，如果是 T_i 是通配符，那么 g_{m-i-1} 为 0，否则为 T_i 的 ASCII 码。
注意这里 T 转化到 g 的顺序是反的。

$$h_i = \sum_{j=0}^{m-1} (g_j - f_{i-j})^2 g_j$$

$$h_i = \sum_{j=0}^{m-1} g_j^3 - 2g_j^2 f_{i-j} + g_j f_{i-j}^2$$

如果 $h_i = 0$，那么说明 $f_i, f_{i-1}, \ldots, f_{i-m+1}$ 和 $g_0, g_1, \ldots, g_{m-1}$ 匹配了。
也就是 $S_{i-m+1}, S_{i-m+2}, \ldots, S_i$ 和 $T_0, T_1, \ldots, T_{m-1}$ 匹配了。
注意 A 和 g 的顺序相反。
因为 g_j^3 是常数，所以不需要 FFT，所以需要 2 次 FFT。

70 bzoj 4259

70.1 题目背景

bzoj 4259 残缺的字符串

70.2 题目大意

输入两个带通配符的字符串 A, B。A 的长度为 m，B 的长度为 n。问 A 在 B 的哪些位置出现了。

70.3 题目分析

将 B 转化为数组 f，如果是 B_i 是通配符，那么 f_i 为 0，否则为 B_i 的 ASCII 码。
将 A 转化为数组 g，如果是 A_i 是通配符，那么 g_{m-i-1} 为 0，否则为 A_i 的 ASCII 码。
注意这里 A 转化到 g 的顺序是反的。

$$h_i = \sum_{j=0}^{m-1} (g_j - f_{i-j})^2 g_j f_{i-j}$$

$$h_i = \sum_{j=0}^{m-1} g_j^3 f_{i-j} - 2g_j^2 f_{i-j}^2 + g_j f_{i-j}^3$$

如果 $h_i = 0$，那么说明 $f_i, f_{i-1}, \ldots, f_{i-m+1}$ 和 $g_0, g_1, \ldots, g_{m-1}$ 匹配了。
也就是 $B_{i-m+1}, B_{i-m+2}, \ldots, B_i$ 和 $A_0, A_1, \ldots, A_{m-1}$ 匹配了。
注意 A 和 g 的顺序相反。
所以需要 3 次 FFT。

71 bzoj 5217

71.1 题目背景

bzoj 5217 航海舰队
题解

71.2 题目大意

所有船要同时移动，相对位置不能改变，不能撞障碍物，不能出界。
问从初始位置，所有船能到达位置的并集是多大?

71.3 题目分析

找到最小的矩形，覆盖所有的船，考虑矩形的左上角，有哪些可行的位置。
相当于是一个二维匹配的问题，船不能匹配到障碍物，这是一个二维 FFT。
除了少部分需要循环卷积的，绝大部分二维 FFT 均可以直接转为一维 FFT。
因为船每步只能上下左右，那么矩形的左上角，也只能上下左右，可以 DFS/BFS 得到一个连通并且可行的左上角位置。
再做一次卷积，计算船可能出现在哪些位置，对于船可能出现的位置，统计个数。

72 bzoj 3527

72.1 题目背景

bzoj 3527 [Zjoi2014] 力

72.2 题目大意

给出 n 个数字 q_i，给出 E_j 的定义如下：

$$E_j = \sum_{i<j} \frac{q_i}{(i-j)^2} - \sum_{i>j} \frac{q_i}{(i-j)^2}$$

求 E_j。

72.3 题目分析

注意到相邻两个 q_i 距离相等，所以相乘时的系数类似于一个卷积的形式。

73 bzoj 3992

73.1 题目背景

bzoj 3992 序列统计
题解

73.2 题目大意

输入一个集合 S，问有多少个长度为 n 的序列，每一项都是 S 中的数字，并且所有项的乘积模 M 为 x。

$M \le 8000$ 且是质数。S 中所有元素非负，且小于 M。$1 \le x \le M-1$。$1 \le n \le 10^9$。

方案数模 1004535809 输出。

73.3 参考题解

S 中的 0 可以忽略。

找到一个 M 的原根 r，把 S 中的每个数字换成以 r 为底的指标。

序列乘积模 M 的限制就变成了，指标之和模 $M-1$ 的限制。

然后就可以 NTT 了。

这个是循环卷积，因为 $M-1$ 的性质并不好，所以必须手动循环卷积。

74 hdu 5958

74.1 题目背景

hdu 5958 New Signal Decomposition

74.2 题目大意

高精度 A * B

74.3 题目分析

直接 FFT。

75 hdu 4609

75.1 题目背景

hdu 1609 3-idiots

bzoj 3513 [MUTC2013]idiots

75.2 题目大意

输入 $n(3 \le n \le 100000)$ 个木棒的长度 $a_i(1 \le a_i \le 100000)$。

从中随机选 3 根，问能组成三角形的概率?

75.3 题目分析

只需要计算无法组成三角形的概率，相当于希望知道，从中选两根不同的，和为 i 的方案数。

对于输入的数列 $\{a_i\}$，构造多项式

$$A(x) = \sum x^{a_i}$$

答案是

$$\frac{A^2(x) - A(x^2)}{2}$$

然后从小到大统计较小的两根的和小于等于较大的一根的情况即可。

76 spoj TSUM

76.1 题目背景

spoj TSUM - Triple Sums

76.2 题目大意

给出数列 $\{a_i\}$，从中取出一个长度等于 3 的子序列，问子序列的和有多少种，对于每种和又有多少个子序列。

$-20000 \le a_i \le 20000$

76.3 题目解法

对于输入的数列 $\{a_i\}$，构造多项式

$$A(x) = \sum x^{a_i}$$

即 x^k 的系数为 k 在 $\{a_i\}$ 中出现了几次。

最终答案的生成函数为

$$\frac{A^3(x)-3A(x)A(x^2)+2A(x^3)}{6}$$

对于 a_i 负数的情况，可以让所有数字都减去最小的 a_i，这样相当于将最后的和都减去了三倍的最小 a_i。

77 bzoj 3771

77.1 题目背景

bzoj 3771 Triple

题解

77.2 题目大意

给出数列 $\{a_i\}$，从中取出一个长度小于等于 3 的子序列，问子序列的和有多少种，对于每种和又有多少个子序列。

$0 \le a_i \le 40000$

77.3 题目分析

对于输入的数列 $\{a_i\}$，构造多项式

$$A(x)=\sum x^{a_i}$$

即 x^k 的系数为 k 在 $\{a_i\}$ 中出现了几次。

最终答案的生成函数为

$$A(x)+\frac{A^2(x)-A(x^2)}{2}+\frac{A^3(x)-3A(x)A(x^2)+2A(x^3)}{6}$$

78 Codechef FARASA

78.1 题目背景

Furik and Rubik and Sub Array

78.2 题目大意

输入长度为 n 的正整数数列 $\{a_i\}$，问所有连续的子序列，有多少种不同的和。

设 a_i 所有数字的和为 S。

数据范围 $1 \le nS \le 4\times 10^{10}$

78.3 题目解法

分类处理。

对于 n 特别小的情况，可以生成所有 n^2 个部分和，然后排序去重。

对于 n 一般小的情况，可以开大小为 S 的 bitset，然后标记。

对于 n 比较大的情况，需要使用 FFT。

设 $s_i(0 \le i \le n)$ 为前缀和，相当于要从 s_i 中选取两个值作差，求绝对值，问有多少个不同的结果。

构造多项式

$$P(x) = \sum x^{s_i}, P'(x) = \sum x^{-s_i}$$

考虑 $P(x)P'(x)$ 中次数大于 0 的部分即可。

事实上，这个题暴力跑的飞快。

79 Codechef COUNTARI

79.1 题目背景

Codechef COUNTARI Arithmetic Progressions

bzoj COUNTARI

79.2 题目大意

输入一个长度为 n 的数组 $\{a_i\}$

问有多少组 $i, j, k(1 \le i < j < k \le n)$ 满足 $a_k - a_j = a_j - a_i$。

$3 \le n \le 100000, 1 \le a_i \le 30000$

79.3 题目解法

设 a_i 的范围为 m。

一个暴力的做法复杂度为 nm，枚举中心点，两侧的值做卷积，但是这个卷积只需要一项。

考虑分块，一块的大小为 T。

所有对可能有两个在同一块里，可能三个在两两不同的块。

对于第一个情况，枚举一块，枚举块内两个下标，时间复杂度为 $\frac{n}{T} \times T^2 = nT$。

对于第二个情况，枚举中间块，两侧的值做卷积，时间复杂度为 $\frac{n}{T} m \log m$。

总复杂度 $nT + \frac{n}{T} m \log m$，所以大概选择 $T = \sqrt{m \log m}$ 最优。

80 Codechef PRIMEDST

80.1 题目背景

Prime Distance On Tree

80.2 题目大意

输入一棵树，边长均为 1，问有多少对点之间的路径长度为质数。

80.3 题目解法

做法昭然可揭，就是点分治，然后每个点 DFS 出所有的深度，自己和自己做卷积（FFT）

80.4 备注

本质上，我们并没用利用质数的任何性质。
也就是说这个题的质数可以改成任何数字。

81 PKU Campus 2014 Problem D

81.1 题目背景

Colorful World

81.2 题目大意

给定 $Ap = q$ 中的 A 和 q，其中 A 是 $n \times n$ 的循环矩阵，p 和 q 是 $n \times 1$ 的矩阵。
所有数均是实数，求解 p。$(n \le 1024)$
循环矩阵即存在长度为 n 的数组 a，满足 $A_{ij} = a_{(i-j) \bmod n}$。

81.3 题目分析

暴力的消元并无法通过。
循环矩阵有什么特殊性质?

81.4 题目解答

考虑三个多项式 $A(x) = \sum_{i=0}^{n-1} a_i x^i$ 和 $P(x) = \sum_{i=0}^{n-1} p_i x^i$ 和 $Q(x) = \sum_{i=0}^{n-1} q_i x^i$。
注意到如果 $x^n = 1$，我们有 $A(x)P(x) = Q(x)$。
（其实就是循环卷积）
而满足 $x^n = 1$ 恰好是 n 个单位根，
我们可以在 n 个单位根计算 $P(\omega_i) = Q(\omega_i)/A(\omega_i)$ 最后用暴力 DFT 算出 p_i 的值。
时间复杂度 $O(n^2)$。
如果 n 是 2 的整数次幂，这个算法可以优化到 $O(n \log n)$。
如果 n 不是 2 的整数次幂，这个算法也可以优化到 $O(n \log n)$。
Chirp-Z 转换

82 bzoj 4002

82.1 题目背景

bzoj 4002 [JLOI2015] 有意义的字符串

82.2 题目大意

输入 b, d, n 求

$$\left\lfloor \left(\frac{b+\sqrt{d}}{2}\right)^n \right\rfloor \bmod 7528443412579576937$$

82.3 参考题解

注意到

$$\left(\frac{b+\sqrt{d}}{2}\right)^n + \left(\frac{b-\sqrt{d}}{2}\right)^n$$

是一个线性递推数列。

并且第二部分绝对值小于 1。

83 ProjectEuler 258

83.1 题目背景

A lagged Fibonacci sequence

83.2 题目大意

定义数列 g_n，如下：

$$g_i = \begin{cases} 1 & 0 \le i \le 1999 \\ g_{i-2000} + g_{i-1999} & i \ge 2000 \end{cases}$$

求 $n = 10^{18}$ 时的 g_n 结果对 20092010 取模。

83.3 题目分析

$O(k^2 \log n)$，可以在时限内出解。

简而言之，就是在模 20092010 的前提下，计算 $x^{10^{18}} \bmod (x^{2000} - x - 1)$ 的结果，设为

$$\sum_{i=0}^{1999} r_i x^i$$

计算 $\sum_{i=0}^{1999} r_i g_i$ 就是答案，相当于直接将 $x = 1$ 带入求值。

其中多项式取模的部分可以有很多种实现方法，比如暴力倍增，或者是用 Sage 之类的语言。

83.4 Cayley–Hamilton 定理

首先我们陈述 Cayley–Hamilton 定理

考虑转移矩阵 M 的特征多项式 $p(x)$，我们有 $p(M)=0$。

在本题中，即有 $M^{2000}-M-I=0$。这个形式非常喵，他表示

$$M^{10^{18}}=f(M)(M^{2000}-M-I)+g(M)$$

其中 $f(M)$ 是一个关于 M 的多项式，相当于商。

$g(M)$ 是一个关于 M 小于 2000 次的多项式，相当于余数。

第一部分一定是 0，我们可以根据第二部分 $g(M)$ 和初始值计算出最终的答案。

83.5 特征多项式

我们设 A 为 $n\times n$ 的方阵，I 是 $n\times n$ 的单位矩阵，那么 A 的特征多项式就是

$$p(\lambda)=\det(\lambda I-A)$$

其中 det 表示行列式求值。

在本题中，特征多项式为 $x^{2000}-x-1$。

84 ProjectEuler 458

84.1 题目背景

Permutations of Project

84.2 题目大意

字符集大小是 7，问有多少个长度为 n 的字符串，7 个字符的任意排列，都不能作为子串出现。

换句话说，任意连续 7 个字符，均不包含全部 7 个字符。

84.3 题目分析

设字符集大小为 k。

设答案为 t_n，除此之外，定义数列 s_n。

s_n 为除了最后 k 个字符之外，均满足条件，并且最后 k 个字符是一个指定的排列。

显然无论指定哪个排列，答案都是一样的。

有两个关系式

$$kt_{n-1}=t_n+k!s_n$$

即对于任意合法方案，再加一个字符之后，可能形成一个新的合法方案，或者是最后 k 个字符构成一个排列。

$$t_n = \sum_{i=1}^{k} (k-i)! s_{n+i}$$

即对于任意合法方案，依次加入 k 个字符，在第 i 个字符加入之后，最后 k 个字符构成一个排列。

这样根据初始值

$$t_i(0 \le i < k) = k^i$$

$$s_i(0 \le i < k) = 0$$

84.4 生成函数

设答案的生成函数为 T。

设最后 7 位是一个排列，并且除了最后 7 位，均满足题目要求的条件的生成函数为 S。

$$1 + 7xT = T + 7!S$$

$$x^7 T = S + 1!xS + 2!x^2 S + \cdots + 6!x^6 S$$

解得

$$T = \frac{1 + x + 2x^2 + 6x^3 + 24x^4 + 120x^5 + 720x^6}{1 - 6x - 5x^2 - 8x^3 - 18x^4 - 48x^5 - 120x^6}$$

85 Codechef DMCS

85.1 题目背景

Dai Ma Chu Shi

85.2 题目大意

用 $1 \times 1 \times 1 \times 1 \times 2$ 的立方体，填满 $2 \times 2 \times 2 \times 2 \times n$ 的空间。

其中立方体是可以旋转的，也就是说他们可以被旋转成 $2\times1\times1\times1\times1$ 或者 $1\times1\times2\times1\times1$。

输入一个 $n(1 \le n \le 10^9)$，求方案数模 $10^9 + 7$ 的结果。

85.3 题目解法

状态压缩 DP，然后一个大小为 2^{16} 的矩阵乘法是显而易见的。

然后我们优化这个矩阵乘法，去掉其中的 0 状态，和对称状态。

这样大概剩下几千个，我们可以预处理前几千项，然后计算线性递推。

具体来说，接口处我们用 1 表示切过一个 $1 \times 1 \times 1 \times 1 \times 2$ 的立方体，用 0 表示没有切过。这样在切口处，一定有偶数个 1。

这可以剪枝掉一大部分。

类似的思路，我们可以对接口黑白染色，像国际象棋棋盘一样，这样在黑色区域内的 1 的个数必须和在白色区域内的 1 的个数相同。

这又可以剪枝掉一大部分。

还有一个情况需要考虑，就是在所有状态中，很多状态可以通过对称和旋转得到，对于这些状态我们只需要记录一次而不需要每次都记录。

这又可以剪枝掉一大部分。

如果我们的做法一样这里我们会得到一个 571 阶的矩阵。

85.4 进一步优化

我们需要考虑两个问题:

1. 已知初始值和转移矩阵，求递推公式?
   - 以此可以使用 $k^2 \log n$ 的优化。
2. 已知初始值和递推公式，求递推公式至少有几阶?
   - 举例，比如初始值是 $1, 1, 2, (3, 5, 8, 13, \ldots)$，递推公式是 $f_i = 2f_{i-2} + f_{i-3}$。
   - 递推公式可以只有 2 阶，也就是 $f_i = f_{i-1} + f_{i-2}$。

已知初始值和转移矩阵，求递推公式

假设初始值是 f，转移矩阵是 A。

那么对于 $f, Af, A^2, \ldots, A^k f$ 这 $k+1$ 个向量来说

存在 $a_i (1 \le i \le k)$ 使得

$$A^k f = \sum_{1 \le i \le k} a_i A^{k-i} f$$

只考虑列向量的第一项

$$f_k = \sum_{1 \le i \le k} a_i f_{k-i}$$

即为递推数列。

已知初始值和递推公式，求递推公式至少有几阶

考虑如下矩阵

$$\begin{bmatrix} f_0 & f_1 & \cdots & f_k \\ f_1 & f_2 & \cdots & f_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ f_k & f_{k+1} & \cdots & f_{2k} \end{bmatrix}$$

这个矩阵的行列式一定是 0，他的秩就是递推数列的阶数。

通过计算他的秩，可以同时得到最简的递推公式表达。

通过这个求秩的优化之后，可以得到一个 71 阶的线性递推。

并使用任意方法（矩阵乘法，或优化后的 $O(k^2 \log n)$ 的做法）通过这个题目。

在出这个题目之初我没有注意到最后一个求秩的优化，所以我误认为我化简出的一个 500 多阶的矩阵是最优解，出了一个愚蠢的数据范围。

86 bzoj 1494

86.1 题目背景

[NOI2007] 生成树计数

86.2 题目大意

n 个点构成一条链，如果两个点之间的距离 $\le k$，那么他们之间有边。
问生成树个数。

86.3 题目解法一

当时的官方解法大概如下。
如果会行列式对质数取模，用高斯消元求值，便可以得到 60 分的高分。
第 7 个点和第 8 个点本意是面相写出 $O(n)$ 状态压缩的选手的，但是实际上行列式写得好也能过。
如果想通过这个题，需要想到状态压缩的做法。
考虑存下最后 k 个点的连通性，对于 $k = 2, 3, 4, 5$ 分别有 $2, 5, 15, 52$ 种可能。
这个是贝尔数，或者说是第二类斯特灵数的求和。
之后转移就枚举下一个点和之前 k 个点的连接情况共 2^k 种可能性。
不能出现环。
如果即将抛弃的点和其他 $k-1$ 个点均不连通，那么必须和新加进来的点连通。
生成转移矩阵，转移即可。

86.4 题目解法二

出题人以为题目中抛出一个看似无用的行列式，可以转移大家焦点，隐藏题目真实解法。但是实际上这个题在近年有了新的做法，行列式的计算复杂度是 $O(n^3)$。
通过行列式可以得到前几百项的结果。
设想这个答案可以通过矩阵乘法求出，那么他必然是一个 ≤ 52 阶的递推数列。
不妨设递推数列为

$$f_n = \sum_{1 \le i \le d} a_i f_{n-i}$$

你可以直接用前 $2n+1$ 项，列出 n 个方程，解出答案，但是略微麻烦。
解方程，除了解出唯一解，还有两个可能，无穷多解和无解。

87 hdu 4914

87.1 题目背景

Linear recursive sequence

87.2 题目大意

模板题。

87.3 题目分析

多项式取模。
倍增。
注意到主要有两部分，多项式乘法，多项式取模。
因为取模的多项式只有三项，所以多项式取模可以暴力。
多项式乘法用 FFT。

88 bzoj 3625

88.1 题目背景

bzoj 3625 小朋友和二叉树
题解

88.2 题目大意

输入一个正整数集合 $\{c_1, c_2, ..., c_n\}$。
二叉树的每个权值只能在这个集合中。
一个二叉树的权值定义为所有点权的权值之和。
对于 $s(1 \le s \le n)$ 计算出有多少个权值为 s 的二叉树，答案对 998244353 取模。

88.3 题目分析

设 g 为 $\{c_i\}$ 的生成函数。
方程为 $f = f^2 g + x^0$，解得

$$f = \frac{1 \pm \sqrt{1-4g}}{2g} = \frac{2}{1 \mp \sqrt{1-4g}}$$

带入 $g(x) = 0$，$f(x)$ 应为 1，所以

$$f = \frac{1 - \sqrt{1-4g}}{2g} = \frac{2}{1 + \sqrt{1-4g}}$$

是最终的答案。特别的当 $g(x) = x$ 时，这个数列是卡特兰数。

89 bzoj 3684

89.1 题目背景

bzoj 3684 大朋友和多叉树

题解

89.2 题目大意

叶子点权为 1，非叶子节点的孩子个数必须是输入的集合 D 中的值。
问有多少个多叉树，恰好包含 s 个叶子节点。

89.3 题目分析

设 $F(x)$ 是答案的生成函数。

$$F(x) = x + \sum_{i \in D} F^i(x)$$

$$x = F(x) - \sum_{i \in D} F^i(x)$$

考虑这样的问题，已知 $G(x) = x - \sum_{i \in D} x^i$。
求多项式 $F(x)$ 满足 $G(F(x)) = x$。
实际上只需要知道 $F(x)$ 的 s 次项系数。

$$[x^s]F(x) = \frac{1}{s}[x^{s-1}](\frac{x}{G(x)})^s$$

右边对 x^s 次方取模。

90 hdu 5730

90.1 题目背景

Shell Necklace

90.2 题目大意

长为 i 的项链有 a_i 种装饰方法，问长度为 n 的项链有多少种装饰方式，求方案数模 313。
$1 \leq n \leq 100000, 0 \leq a_i \leq 10000000$

90.3 题目分析

列出动态规划方程 $f_i = \sum_{1 \leq j \leq i} a_j f_{i-j}$
这个题目有两种做法。
一种是喜闻乐见的 CDQ 分治。
另一种是直接认为负数部分 $f_i = 0 (i < 0)$，然后进行多项式取模。

91 FWT/FMT

介绍和位运算相关的 FWT 和 FMT。

炫酷反演魔术

91.1 FWT

FWT 从过程上来说，就是 k 维，每一维度长度为 2 的 FFT。

这个过程本身是 $O(n^2) = O(4^k)$ 的复杂度，然后可以优化成 $O(k2^k)$。

91.2 FMT

FMT 从过程上来说，就是 k 维，每一维度长度为 2 的部分和。

这个过程本身是 $O(n^2) = O(4^k)$ 的容斥原理的复杂度，然后可以每一维度分别求部分和优化成 $O(k2^k)$。

92 Codeforces 453D

92.1 题目背景

Little Pony and Elements of Harmony

92.2 题目大意

也是一个异或的转移。

92.3 题目分析

裸的 FWT，直接做就可以了。

定义新数组 c[i] 为 b[count(i)]，即数一下 i 中有多少个 1，把对应的 b[count(i)] 赋值给 c[i]。

然后对初始值和数组 c[i] 进行异或卷积即可，分别 FWT，然后 c[i] 变成 t 次方，然后乘起来，FWT 回去。

然后本题模的数 p 不是质数，需要通过中间模 np 解决。

93 bzoj 4589

93.1 题目背景

Hard Nim

93.2 题目大意

输入 n, m，选出 n 个 m 以内的质数，要求异或为 0，问有多少组解。

93.3 题目分析

相当于问有多少组方案，选出来 n 个质数，使得他们的异或为 0。

考虑一个时间复杂度是 $O(nm^2)$ 的动态规划。

显然是存在一个 $O(nm^2)$ 注意到转移是一个类似于 FWT 的形式。

FWT 可以用快速幂优化，这个就是一个弱智题了。

随便翻到的题解

94 bzoj 4036

94.1 题目背景

bzoj 4036: [HAOI2015] 按位或

94.2 题目大意

一共有 n 个球，初始都是白色，每次随机一个子集，把子集中的球染黑，问期望多少次操作之后所有球都变黑了。

94.3 题目解答

Maximum-minimums identity

相当于要求所有位置变成 1 的时间的最大值的期望。

相当于要求所有位置的子集变成 1 的时间的最小值的期望。

对于子集 S 最小值的期望来说，他的值等于 $\frac{1}{1-p}$，其中 p 是操作一次，操作不到这个集合的概率

也就是 S 补集的所有子集对应的概率之和。

假设 p 为 1，那么无解。

94.4 参考题解

参考代码

95 hdu 5909

95.1 题目背景

Tree Cutting

95.2 题目大意

给出一棵 n 个点的树，有点权 $v_i(0 \le v_i < m)$，求异或和为 $x(0 \le x < m)$ 的非空连通子图的个数。

方案数模 1000000007。

95.3 题目分析

每个点可以写出一个 DP 方程，时间复杂度是 $O(m^2)$。
方程显然可以用 FWT 优化，变成了 $O(m\log m)$。
总复杂度 $O(nm\log m)$。
注意到可以最后一起 DFT，总复杂度 $O(nm+m\log m)$。

96 uoj 272

96.1 题目背景

石家庄的工人阶级队伍比较坚强
当时比较年轻，还不懂得世事的艰辛。

96.2 题目大意

96.3 题目分析

三进制的 FWT。
要点在于对于这个取模的数字，一定有三次单位根，可以通过原根计算。
一定要和 CF 453D 一样构造出两个多项式，再做变换。
而不要直觉去猜每个系数怎么变化的，非常麻烦。

97 hdu 4626

97.1 题目背景

hdu 4626 Jinkeloid

97.2 题目大意

97.3 题目分析

计算前缀中每个字母出现次数的奇偶性。
对于输入的 k 位，枚举 0 到 2^k-1 的每个数字。
对于这些位置上的数字，必须是确定的 0 和 1，其他位置是 0 和 1 都可以。
这可以用子集和变换，做的时候再反向子集和变换一次，算组合数。

98 hdu 5823

98.1 题目背景

hdu 5823 color II

98.2 题目大意

98.3 题目分析

设 $f[i][S]$ 为用 i 种颜色，染集合 S 的方案数。（可行性）

如果 S 是独立集，那么 $f[1][S]=1$。

对于其他情况

$$f[i][S]=\sum_{u \text{ xor } v=S} f[1][u]\times f[i-1][v]$$

注意到如果 u 和 v 有交集，那么一定不是最优解，不影响答案。

最后考虑 $f[i][S]$ 是否为零即可。

99 hdu 6057

99.1 题目背景

hdu 6057 Kanade's convolution

99.2 题目大意

输入两个长度为 2^m 的序列 $A[i],B[i]$。

$$C[k]=\sum_{i \text{ bitand } j} A[i \text{ xor } j]\times B[i \text{ bitor } j]$$

求数列 $C[i]$。

99.3 题目分析

设 $x=i \text{ bitor } j, y=i \text{ xor } j$ 在 x,y 确定的情况下，(i,j) 解的个数是 $2^{\text{count}(y)}$。

$$C[k]=\sum_{x \text{ xor } y=k}[\text{count}(x)-\text{count}(y)=\text{count}(k)]B[x](A[y]2^{\text{count}(y)})$$

可以枚举 $\text{count}(x)$ 和 $\text{count}(y)$，然后把符合条件的 $B[x]$ 和 $(A[y]2^{\text{count}(y)})$ 筛选出来，做异或卷积，结果放到 $\text{count}(k)$。最后统计答案即可。

大概需要做 m 次 FWT，然后中间的乘法需要 $m^2 2^m$ 次，最后变换回去需要一次 FWT。

总复杂度 $m^2 2^m$。

100 AtCoder Regular Contest 100 Problem E

100.1 题目背景

AtCoder Regular Contest 100 Problem E

100.2 题目大意

和以往的求每个子集的和不同。
这次需要求每个子集中，最大值和次大值。

100.3 题目解法

直接子集和变换。
加法变成 4 个数字选 2 个最大的。

100.4 参考代码

我的做法

101 CSAcademy Maxor

101.1 题目背景

Maxor

101.2 题目大意

输入 $n(2 \le n \le 10^5)$ 个数字 $a_i(0 \le a_i < 2^{17})$，从中选出 2 个，使得位或结果最大。

101.3 题目解法

统计个数，位或卷积，找到第一个次数不为 0 的。

102 Codeforces 210418B3

102.1 题目背景

Two Swords Style Hard
感谢 dreamoon。

102.2 题目大意

有 20 种属性，n 把剑，每把剑可能具有，或者不具有每种属性。
完成任务可以携带两把剑，获得这两把剑属性的并集。
完成每个任务有属性的要求，获得的属性，必须是要求的属性的超集才可以。

102.3 题目分析

三次子集和变换。
需要翻墙的题解

103 群论

在数学中，群是由一个集合以及一个二元运算所组成的，符合下述四个性质（称为“群公理”）的代数结构。这四个性质是封闭性、结合律、单位元和对于集合中所有元素存在逆元素。

首先我们有一个集合 S。

1. 封闭性：对于所有 G 中的 a, b，运算 $a \times b$ 的结果也在 G 中。
2. 结合律：对于所有 G 中的 $a, b,$ 和 c，等式 $(a \times b) \times c = a \times (b \times c)$ 成立。
3. 单位元：存在 G 中的一个元素 e，使得对于所有 G 中的元素 a，总有等式 $e \times a = a \times e = a$ 成立。
4. 逆元：对于每个 G 中的 a，存在 G 中的一个元素 b 使得总有 $a \times b = b \times a = e$，此处 e 为单位元。

群运算的次序很重要，把元素 a 与元素 b 结合，所得到的结果不一定与把元素 b 与元素 a 结合相同；亦即 $a \times b = b \times a$（交换律）不一定恒成立。满足交换律成立的群称为交换群（阿贝尔群，以尼尔斯・阿贝尔命名），不满足交换律的群称为非交换群（非阿贝尔群）。

103.1 提出问题

以一个环为例，我们希望求将他染色之后有多少种不同的方案。

我们认为旋转之后是相同的。

旋转操作可以认为是一个 1 到 n 的排列。

所有这样的操作构成了一个群。

103.2 用术语表述

如果两种染色方案，经过变换可以变成相同的，称之为在一个轨道上。

所有可以变成相同的，均在这个轨道上。

我们所需要的答案，就是**轨道数**；即有多少种方案经过变换之后也不相同。

103.3 Burnside 引理

Burnside 引理

$$|X/G| = \frac{1}{|G|} \sum_{g \in G} |X^g|$$

103.4 Polya 定理

Polya 定理是 Burnside 引理在染色情况下的一个特例。

我们可以把排列，写成若干个轮换的形式，其中轮换的个数叫做**循环指标**。

103.5 一维的情况

答案是

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

如果有对称的话，需要根据 n 的奇偶进行讨论。如果 n 是奇数

$$\frac{1}{2n}(nc^{(n+1)/2}+\sum_{i=1}^{n} c^{\gcd(i,n)})$$

如果 n 是偶数

$$\frac{1}{2n}(n/2c^{n/2}+n/2c^{n/2+1}+\sum_{i=1}^{n} c^{\gcd(i,n)})$$

其中 $\sum_{i=1}^{n} c^{\gcd(i,n)}$ 部分可以进一步优化为先枚举最大公约数。

$$\sum_{i=1}^{n} c^{\gcd(i,n)} = \sum_{i|n} c^{i}\varphi(n/i) = \sum_{i|n} c^{n/i}\varphi(i)$$

所以只需要预处理 n 的所有约数 x 和对应的欧拉函数值 $\varphi(x)$ 即可。

103.6 二维的情况

答案是

$$\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} c^{nm/\operatorname{lcm}(n/\gcd(i,n),m/\gcd(i,m))}$$

同样可以进一步优化为先枚举最大公约数，设 $a=n/\gcd(i,n), b=m/\gcd(j,m)$ 原式可以化为

$$\frac{1}{nm}\sum_{a|n}\sum_{a|m}\varphi(a)\varphi(b)c^{nm/\operatorname{lcm}(a,b)}$$

所以只需要预处理 n,m 的所有约数 x 和对应的欧拉函数值 $\varphi(x)$ 即可。

103.7 生成数据

https://gist.github.com/dario2994/fb4713f252ca86c1254d
在 ‘int‘范围内比较大的高合成数（任何比它小的自然数的约数个数均比这个数的约数个数少）
$735134400 = 2^6 * 3^3 * 5^2 * 7 * 11 * 13 * 17$，1344 个约数
$1102701600 = 2^5 * 3^4 * 5^2 * 7 * 11 * 13 * 17$，1440 个约数
$1396755360 = 2^5 * 3^3 * 5 * 7 * 11 * 13 * 17 * 19$，1536 个约数
$2095133040 = 2^4 * 3^4 * 5 * 7 * 11 * 13 * 17 * 19$，1600 个约数

103.8 全排列/自然数拆分

一类题目需要枚举全排列，以最经典的 sgu282 为例子。

104 poj 2409

104.1 题目背景

poj 2409 Let it Bead

104.2 题目大意

用 m 种颜色给长度为 n 的环染色，旋转和翻转相同算同一种方案，问有多少种不同的方案。

104.3 题目分析

旋转部分枚举 gcd。
翻转部分按奇偶讨论。

105 poj 2888

105.1 题目背景

poj 2888 Magic Bracelet

105.2 题目大意

用 m 种颜色给一串长度为 n 的项链染色，其中有 k 对颜色不能相邻，旋转相同看作同一种方案，问方案数，结果模 9973。

105.3 题目分析

考虑没有旋转怎么做，需要矩阵乘法。
有旋转的之后就是枚举 gcd。

106 hdu 5868

106.1 题目背景

hdu 5868 Different Circle Permutation

106.2 题目大意

用黑白两种颜色给一串长度为 n 的项链染色，白色不能相邻，旋转相同看作同一种方案，问方案数，结果模 1000000007。

106.3 题目分析

考虑没有旋转怎么做，需要矩阵乘法。（Lucas Number）
有旋转的之后就是枚举 gcd。

107 bzoj 1004

107.1 题目背景

bzoj 1004 [HNOI2008]Cards

107.2 题目大意

输入一个置换群。
问有多少种不同染色方案在这个置换群下是等价的。

107.3 题目分析

在一个轮换中的颜色必须相同。
做一个类似背包的 DP 即可。

107.4 数据太弱了

看到这份题解后，我去检查了一下。
确实输入的排列对于答案都没有任何贡献，直接除以 $m+1$ 即可。

108 sgu 282

108.1 题目背景

sgu 282
非常好的 Polya 题目。

108.2 题目大意

一个 n 个点的完全无向图，对每个边染 m 色之一。
问有多少种本质不同的染色方案。

108.3 题目分析

枚举所有排列，对于每个排列计算答案。
可以注意到只有轮换的长度才会影响答案。
所有边可以分为两种情况，一个环内部的，两个环之间的。
一个环内部的是环长除以 2 下取整。
两个环之间的是两个环长的最大公约数。

108.4 参考资料

算法合集之《Polya 计数法的应用》

109 bzoj 1488

109.1 题目背景

bzoj 1488 [HNOI2009] 图的同构

109.2 题目大意

求两两互不同构的含 $n(n \leq 60)$ 个点的简单图有多少种。
简单图是关联一对顶点的无向边不多于一条的不含自环的图。
可以不连通，答案对 997 取模。

109.3 题目分析

sgu282 的削弱版。