



不定方程与同余方程组

前置芝士

exgcd求解不定方程 $ax+by=\gcd(a,b)$ / 线性同余方程 $ax\equiv b(\text{mod } m)$ 的解

exgcd求解不定方程 $ax+by=\gcd(a,b)$

设

$$ax_1 + by_1 = \gcd(a, b)$$

$$bx_2 + (a\%b)y_2 = \gcd(b, a\%b)$$

由欧几里得定理可得

$$\gcd(a, b) = \gcd(b, a\%b)$$

于是

$$ax_1 + by_1 = bx_2 + (a\%b)y_2$$

$$ax_1 + by_1 = bx_2 + (a - \lfloor a/b \rfloor * b)y_2$$

整理

$$ax_1 + by_1 = ay_2 + b(x_2 - \lfloor a/b \rfloor * y_2)$$

于是

$$x_1 = y_2$$

$$y_1 = x_2 - \lfloor a/b \rfloor * y_2$$

$ax+by=\gcd(a,b)$ 即可通过最初的 x,y 求解

于是我们可以通过递归求解

```

int exgcd(int a,int b,int &x,int &y)//扩展欧几里得
{
    if(b==0)
    {
        //从x=1,y=0开始向上
        x=1;y=0;
        //ax+by=gcd(a,b)->ax=0->x=1,y=0
        return a;
    }
    //先求解gcd(a,b)
    int d=exgcd(b,a%b,x,y),t=x;
    x=y;y=t-a/b*y;
    return d;
}

```

exgcd求解线性同余方程 $ax \equiv b \pmod{m}$ 的解

$$ax \equiv b \pmod{m}$$

可写成

$$ax + mk = b$$

于是我们先求解不定方程

$$ax + mk = \gcd(a, m)$$

若 $\gcd(a, m) \neq 1$ 则无解，否则得到解

$$x = x_0$$

$$k = k_0$$

于是我们得到原方程的解为

$$x_1 = x_0 * b / \gcd(a, m)$$

$$k_1 = k_0 * b / \gcd(a, m)$$

方程的任意解(对任意整数t成立)为

$$x = x_1 + mt$$

$$k = k_1 - at$$

求最小的正整数解

$$x = (x_1 \bmod t + t) \bmod t$$

其中

$$t = m / \gcd(a, m)$$

要用exgcd求解逆元的话，需要保证gcd(a,m)=1

代入exgcd(a,m,x,y)中,对x值域变换即可

其实就是

$$ax \equiv 1(\bmod m)$$

可写成

$$ax + mk = \gcd(a, m) = 1$$

罢了

中国剩余定理

求解同余方程组

$$\begin{cases} x \equiv a_1(\bmod m_1) \\ x \equiv a_2(\bmod m_2) \\ \dots \\ x \equiv a_k(\bmod m_k) \end{cases}$$

其中

m_1, m_2, \dots, m_k 两两互质

过程:

求

$$M = m_1 * m_2 * \dots * m_k$$

对每个 m_i 求

$$M_i = M / m_i$$

$$M_i^{-1} \equiv 1 \pmod{m_i}$$

$$c_i = M_i^{-1} * M_i$$

于是

$$x = \sum_{i=1}^k a_i * c_i \pmod{M}$$

很显然的证明，对任意一个方程组：

$$x \equiv \sum_{i=1}^k a_i * c_i \pmod{m_i}$$

$$x \equiv a_i * M_i * M_i^{-1} \pmod{m_i}$$

$$x \equiv a_i \pmod{m_i} * (M_i * M_i^{-1} \pmod{m_i})$$

按定义

$$x \equiv a_i \pmod{m_i}$$

代码：

```

int CRT()
{
    int mul=accumulate(m.begin(),m.end(),1LL,
    [](int a,int b){return a*b;}),ans=0;
    for(int i=0;i<n;i++)
    {
        int M=mul/m[i],b,y;
        exgcd(M,m[i],b,y); //求M的逆元
        ans=(ans+nums[i]*M%mul*b%mul+mul)%mul;
    }
    return (ans%mul+mul)%mul;
}

```

扩展中国剩余定理

求解同余方程组

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \dots \\ x \equiv a_k \pmod{m_k} \end{cases}$$

其中

m_1, m_2, \dots, m_k 不两两互质

过程:

考虑合并两个同余方程

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \end{cases}$$

可写成不定方程

$$\begin{cases} x = a_1 + k_1 * m_1 \\ x = a_2 + k_2 * m_2 \end{cases}$$

消去x

$$a_1 + k_1 * m_1 = a_2 + k_2 * m_2$$

于是我们得到了一个不定方程

$$k_1 * m_1 + -k_2 * m_2 = a_2 - a_1$$

可通过exgcd求解

$$K_1 * m_1 + -K_2 * m_2 = gcd(m_1, m_2)$$

于是

$$k_1 = \frac{a_2 - a_1}{gcd(m_1, m_2)} * K_1$$

$$k_2 = \frac{a_1 - a_2}{gcd(m_1, m_2)} * K_2$$

得到x的一个解

$$x_0 = a_1 + k_1 * m_1 = a_1 + \frac{a_2 - a_1}{gcd(m_1, m_2)} * K_1 * m_1$$

窝们很显然可以构造x的通解

$$x = x_0 + t * lcm(m_1, m_2)$$

于是进行形式转化

$$x \equiv x_0(mod\ lcm(m_1, m_2))$$

于是我们得到了两个同余方程的合并

```

int _exCRT()
{
    int M=m[0],ans=nums[0];
    //M: 合并后的模数, ans:合并后的余数
    for(int i=1;i<n;i++)
    {
        //当前方程:
        //x≡nums[i] (mod \ m[i])
        //x≡ans (mod \ M)
        //不定方程 ax+by=gcd(a,b)
        int a=M,b=m[i];
        int c=((nums[i]-ans)%b+b)%b;
        int x,y;
        int gcd=exgcd(a,b,x,y);
        int bg=b/gcd;
        if(c%gcd!=0) return -1;//判断有无解
        x=(x%bg+bg)%bg;//对x值域变换变成正数
        x=(x*c/gcd%bg+bg)%bg;//对x值域变换
        ans+=x*M;
        M*=bg;//更新M=lcm(M,m[i])=m[i]*M/gcd(M,m[i])
        ans=(ans%M+M)%M;
    }
    return (ans%M+M)%M;
}

```

筛法

欧拉函数的定义

$1 \sim n$ 中与 n 互质的数的个数称为欧拉函数, 记为 $\varphi(n)$

例: $\varphi(1) = 1, \varphi(2) = 1, \varphi(3) = 2, \varphi(4) = 2, \varphi(5) = 4$

欧拉函数的性质

1. 若 p 是质数, 则 $\varphi(p) = p - 1$
2. 若 p 是质数, 则 $\varphi(p^k) = (p - 1)p^{k-1}$

3. **积性函数**: 若 $\gcd(m, n) = 1$, 则 $\varphi(mn) = \varphi(m)\varphi(n)$

欧拉函数的计算公式

由唯一分解定理 $n = \prod_{i=1}^s p_i^{\alpha_i} = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_s^{\alpha_s}$,

$$\begin{aligned}\varphi(n) &= \prod_{i=1}^s \varphi(p_i^{\alpha_i}) \\&= \prod_{i=1}^s p_i^{\alpha_i-1} (p_i - 1) \\&= \prod_{i=1}^s p_i^{\alpha_i} \left(1 - \frac{1}{p_i}\right) \\&= \left(\prod_{i=1}^s p_i^{\alpha_i}\right) \times \left(\prod_{i=1}^s \left(1 - \frac{1}{p_i}\right)\right) \\&= n \times \prod_{i=1}^s \frac{p_i - 1}{p_i} \\&= n \times \frac{p_1 - 1}{p_1} \times \frac{p_2 - 1}{p_2} \times \cdots \times \frac{p_s - 1}{p_s}\end{aligned}$$

欧拉函数仅由 n 和质因子决定, 与次数无关。

例: $\varphi(12) = 12 \times \frac{2-1}{2} \times \frac{3-1}{3} = 4$

筛法求欧拉函数

若 i 是质数, $\varphi[i] = i - 1$ 。

在线性筛中, 每个合数 m 都是被最小的质因子筛掉的。

设 p_j 是 m 的最小质因子, 则 m 通过 $m = p_j \times i$ 筛掉。

分两种情况计算:

1. 若 i 能被 p_j 整除 (即 $i \equiv 0 \pmod{p_j}$), 则 i 包含了 m 的所有质因子:

$$\begin{aligned}
 \varphi(m) &= m \times \prod_{k=1}^s \frac{p_k - 1}{p_k} \\
 &= p_j \times i \times \prod_{k=1}^s \frac{p_k - 1}{p_k} \\
 &= p_j \times \varphi(i)
 \end{aligned}$$

例： $\varphi(12) = \varphi(2 \times 6) = 2 \times \varphi(6)$

2. **若 i 不能被 p_j 整除** (即 $\gcd(i, p_j) = 1$) , 则 i 和 p_j 互质:

$$\begin{aligned}
 \varphi(m) &= \varphi(p_j \times i) \\
 &= \varphi(p_j) \times \varphi(i) \\
 &= (p_j - 1) \times \varphi(i)
 \end{aligned}$$

例： $\varphi(75) = \varphi(3 \times 25) = (3 - 1) \times \varphi(25)$

```

vector<int> euler()
{
    vector<int> phi(n+1);
    phi[1]=1;
    vector<int> primes;
    vector<bool> v(n+1,0);
    for(int i=2;i<=n;i++)
    {
        if(!v[i])primes.push_back(i),phi[i]=i-1;
        for(int j=0;j<primes.size()&&primes[j]*i<=n;j++)
        {
            int m=primes[j]*i;
            v[m]=1;
            if(i%primes[j]==0)
            {
                phi[m]=phi[i]*primes[j];
                break;
            }
            else phi[m]=phi[i]*(primes[j]-1);
        }
    }
}

```

筛法求约数个数

问题

给定整数 n ($n \leq 10^6$), 输出 $1 \sim n$ 中每个数的约数个数。

约数个数定理

若正整数 n 有质因数分解 $n = \prod_{i=1}^s p_i^{\alpha_i}$, 则约数个数为:

$$d(n) = \prod_{i=1}^s (\alpha_i + 1)$$

证明

- 对每个质因子 $p_i^{\alpha_i}$, 其约数可取 $p_i^0, p_i^1, \dots, p_i^{\alpha_i}$ 共 $(\alpha_i + 1)$ 种选择
- 根据乘法原理, 总约数个数为各质因子选择数的乘积:

$$d(n) = (\alpha_1 + 1) \times (\alpha_2 + 1) \times \dots \times (\alpha_s + 1)$$

筛法求约数个数

记 $a[i]$ 为 i 的最小质因子的次数, $d[i]$ 为 i 的约数个数。

若 i 是质数,

$$a[i] = 1, \quad d[i] = 2$$

在线性筛中, 每个合数 m 都是被最小的质因子筛掉的。

设 p_j 是 m 的最小质因子, 则 m 通过 $m = p_j \times i$ 筛掉。

(1) 若 i 能被 p_j 整除, 则 p_j 一定是 i 的最小质因子。

$$a[m] = a[i] + 1;$$

$$d[i] = (a[i] + 1) \times \dots, \quad d[m] = (a[m] + 1) \times \dots$$

于是

$$d[m] = d[i] \times \frac{a[m] + 1}{a[i] + 1}$$

(2) 若 i 不能被 p_j 整除, 则 i 不包含质因子 p_j 。

$$a[m] = 1, \quad d[m] = d[i] \times (1 + 1)$$

```

//O(n)求1-n的约数个数
vector<int> d()
{
    vector<int> a(n+1),d(n+1);
    vector<int> primes;
    vector<bool> v(n+1,0);
    for(int i=2;i<=n;i++)
    {
        if(!v[i])
        {
            primes.push_back(i);
            a[i]=1,d[i]=2;
        }
        for(int j=0;j<primes.size()&&primes[j]*i<=n;j++)
        {
            int m=primes[j]*i;
            v[m]=1;
            if(i%primes[j]==0)
            {
                a[m]=a[i]+1;
                d[m]=d[i]/(a[i]+1)*(a[m]+1);
                break;
            }
            else
            {
                a[m]=1;
                d[m]=d[i]*2;
            }
        }
    }
}

```

约数和定理

若 $n = \prod_{i=1}^s p_i^{\alpha_i}$, 则 $f(n) = \prod_{i=1}^s \sum_{j=0}^{\alpha_i} p_i^j$

证明:

$p_i^{\alpha_i}$ 的约数有 $p_i^0, p_i^1, \dots, p_i^{\alpha_i}$ 共 $(\alpha_i + 1)$ 个, 其约数和为 $\sum_{j=0}^{\alpha_i} p_i^j$.

根据乘法原理,

$$f(n) = \prod_{i=1}^s \sum_{j=0}^{\alpha_i} p_i^j$$

例:

$$12 = 2^2 \times 3^1,$$

$$f(12) = (1 + 2 + 4) \times (1 + 3) = 7 \times 4 = 28$$

筛法求约数和

记 $g[i]$ 为 i 的最小质因子的幂和 $1 + p^1 + p^2 + \dots + p^k$, $f[i]$ 为 i 的约数和。

若 i 是质数,

$$g[i] = f[i] = i + 1$$

在线性筛中, 每个合数 m 都是被最小的质因子筛掉的。设 p_j 是 m 的最小质因子, 则 m 通过 $m = i \times p_j$ 筛掉。

(1) 若 i 能被 p_j 整除, 则 p_j 一定也是 i 的最小质因子

$$g[i] = p_j^0 + p_j^1 + \dots + p_j^{\alpha_j}, \quad g[m] = p_j^0 + p_j^1 + \dots + p_j^{\alpha_j+1}$$

$$f[i] = g[i] \times \dots, \quad f[m] = g[m] \times \dots$$

于是

$$f[m] = f[i] \times \frac{g[m]}{g[i]}$$

(2) 若 i 不能被 p_j 整除, 则 i 不包含质因子 p_j 。

$$g[m] = 1 + p_j$$

$$f[m] = g[m] \times f[i]$$

```
//O(n)求1-n的约数和
vector<int> sumd()
{
    vector<int> g(n+1),f(n+1);
    vector<int> primes;
    vector<bool> v(n+1,0);
    g[1]=f[1]=1;
    for(int i=2;i<=n;i++)
    {
        if(!v[i])
        {
            primes.push_back(i);
            f[i]=g[i]=i+1;
        }
        for(int j=0;j<primes.size()&&primes[j]*i<=n;j++)
        {
            int m=primes[j]*i;
            v[m]=1;
            if(i%primes[j]==0)
            {
                g[m]=g[i]*primes[j]+1;
                f[m]=f[i]*g[m]/g[i];
                break;
            }
            else
            {
                g[m]=primes[j]+1;
                f[m]=f[i]*g[m];
            }
        }
    }
}
```

唯一分解定理

$$n = \prod_{i=1}^s p_i^{\alpha_i} = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_s^{\alpha_s}$$

莫比乌斯函数定义

莫比乌斯函数记作 $\mu(n)$ ，它是一个经典的数论函数，定义如下：

- $\mu(1) = 1$
- 如果 n 含有平方因子（即存在某个质数 p ，使得 $p^2 \mid n$ ），则 $\mu(n) = 0$
- 如果 n 是 k 个互不相同的质数的乘积（即 $n = p_1 p_2 \cdots p_k$ ），则：

$$\mu(n) = (-1)^k$$

筛法求莫比乌斯函数

若 i 是质数， $\mu[i] = -1$ 。

在线性筛中，每个合数 m 都是被最小的质因子筛掉的。

设 p_j 是 m 的最小质因子，则 m 通过 $m = i \times p_j$ 筛掉。

(1) 若 i 能被 p_j 整除，则 i 也包含质因子 p_j 。

$$\mu[m] = 0$$

(2) 若 i 不能被 p_j 整除，则 m 比 i 多一个不同的质因子 p_j

- 若 $\mu[i] = -1$ ，则 $\mu[m] = 1$
 - 若 $\mu[i] = 1$ ，则 $\mu[m] = -1$
 - 若 $\mu[i] = 0$ ，则 $\mu[m] = 0$
- 综上， $\mu[m] = -\mu[i]$ 。

线性逆元

$O(n)$ 求阶乘和阶乘逆元

📖 推导目标

给定质数 p ，我们希望在线性时间内计算 1 到 n 的所有数在模 p 意义下的乘法逆元，即：

求 $\forall 1 \leq i \leq n$, 使得 $r_i \cdot i \equiv 1 \pmod{p}$ 的 r_i

💡 推导公式

我们设 $r_i = i^{-1} \pmod{p}$, 有：

- $r_1 = 1$
- 对于 $i > 1$, 我们可以利用如下递推式求出 r_i :

$$r_i = (p - \left\lfloor \frac{p}{i} \right\rfloor) \cdot r_{p \bmod i} \pmod{p}$$

📎 证明过程

考虑：

$$p = i \cdot \left\lfloor \frac{p}{i} \right\rfloor + (p \bmod i) \Rightarrow p \bmod i = p - i \cdot \left\lfloor \frac{p}{i} \right\rfloor$$

两边模 p :

$$i \cdot \left(-\left\lfloor \frac{p}{i} \right\rfloor \right) \equiv -(p \bmod i) \pmod{p} \Rightarrow i \cdot \left(-\left\lfloor \frac{p}{i} \right\rfloor \right) \cdot (p \bmod i)^{-1} \equiv -1 \pmod{p}$$

两边都乘上 -1 :

$$i \cdot \left(\left\lfloor \frac{p}{i} \right\rfloor \right) \cdot (p \bmod i)^{-1} \equiv 1 \pmod{p}$$

于是我们得出：

$$\text{inv}[i] \equiv - \left(\left\lfloor \frac{p}{i} \right\rfloor \right) \cdot \text{inv}[p \bmod i] \pmod{p}$$

再化简成**无负数形式**：

$$\text{inv}[i] = (p - p/i) \cdot \text{inv}[p \% i] \bmod p$$

这就是我们要用的递推式！

◇ C++ 实现示例

```
void preC()
{
    inv[1]=1;
    for(int i=2;i<=n;i++)
    {
        inv[i]=(mod-mod/i)*inv[mod%i]%mod;
    }
    fac[0]=invfac[0]=1;
    for(int i=1;i<=n;i++)
    {
        fac[i]=fac[i-1]*i%mod;
        invfac[i]=invfac[i-1]*inv[i]%mod;
    }
}
```

◇ 时间复杂度

- 时间： $\mathcal{O}(n)$
- 空间： $\mathcal{O}(n)$
- 要求 p 是质数（否则不存在乘法逆元）

和式变换

和式变换规则与技术

基本变换规则

1. 分配律

$$\sum_{k \in K} ca_k = c \sum_{k \in K} a_k$$

2. 结合律

$$\sum_{k \in K} (a_k + b_k) = \sum_{k \in K} a_k + \sum_{k \in K} b_k$$

3. 交换律

$$\sum_{k \in K} a_k = \sum_{p(k) \in K} a_{p(k)}$$

其中 $p(k)$ 是指标集的任意排列

示例:

$$a_1 + a_2 + a_3 + a_6 = a_6 + a_3 + a_2 + a_1$$

高级变换技术

1. 替换条件式

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

2. 替换指标变量

$$\sum_{i=1}^n \sum_{j=1}^m [\gcd(i, j) = k] = \sum_{i'=1}^{\lfloor n/k \rfloor} \sum_{j'=1}^{\lfloor m/k \rfloor} [\gcd(i', j') = 1]$$

其中 $i' = i/k, j' = j/k$

3. 交换求和次序

$$\sum_{i=1}^n \sum_{j=1}^m A(i)B(j) = \sum_{j=1}^m \sum_{i=1}^n A(i)B(j)$$

4. 分离变量

$$\sum_{i=1}^n \sum_{j=1}^m A(i)B(j) = \left(\sum_{i=1}^n A(i) \right) \left(\sum_{j=1}^m B(j) \right)$$

技巧

1. 区间整除条件式的封闭形式

$$\sum_{i=1}^n [k|i] = \left\lfloor \frac{n}{k} \right\rfloor$$

扩展

$$\left\lfloor \frac{\left\lfloor \frac{n}{k} \right\rfloor}{m} \right\rfloor = \left\lfloor \frac{n}{km} \right\rfloor$$

人话：在1到n的整数中，能被k整除的数的个数是 $\left\lfloor \frac{n}{k} \right\rfloor$

2. $[\gcd(i, j) = 1]$ 的进一步变换

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

$$= \sum_{i=1}^n \sum_{j=1}^m \sum_{d=1}^{\min(n, m)} \mu(d) [d|i] [d|j]$$

人话：变换枚举顺序，d能整除i和j，则d能整除gcd(i,j)

3. 有序对和无序对求和的互相转换

$$\sum_{i=1}^n \sum_{j=1}^n A(i, j) = 2 * \sum_{i=1}^n \sum_{j=i}^n A(i, j) - \sum_{i=1}^n A(i, i)$$

当且仅当 $A(i, j) = A(j, i)$ 时成立

人话：将有序对转换为无序对，注意枚举顺序

4. 通过 $\gcd(i, j) = 1$ 构造条件式 $[\gcd(i, j) = 1]$

令

$$d = \gcd(i, j), i = i'd, j = j'd$$

注意，当且仅当

$$\gcd(i', j') = 1$$

时，此变换成立，于是

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

$$= \sum_{d=1}^{\min(n,m)} \sum_{i'd=1}^n \sum_{j'd=1}^m f(d)[\gcd(i',j')=1]$$

$$= \sum_{d=1}^{\min(n,m)} \sum_{i'=1}^{\lfloor n/d \rfloor} \sum_{j'=1}^{\lfloor m/d \rfloor} f(d)[\gcd(i',j')=1]$$

变量换名

$$= \sum_{d=1}^{\min(n,m)} \sum_{i=1}^{\lfloor n/d \rfloor} \sum_{j=1}^{\lfloor m/d \rfloor} f(d)[\gcd(i,j)=1]$$

人话：没有人话

生成函数

序列 a 的普通生成函数 (ordinary generating function, OGF) 定义为形式幂级数 (其实就是一个多项式()) :

$$F(x) = \sum_n a_n x^n$$

a 既可以是有穷序列, 也可以是无穷序列。常见的例子 (假设 a 以 0 为起点) :

1. 序列 $a = \langle 1, 2, 3 \rangle$ 的普通生成函数是 $1 + 2x + 3x^2$ 。
2. 序列 $a = \langle 1, 1, 1, \dots \rangle$ 的普通生成函数是 $\sum_{n \geq 0} x^n$ 。
3. 序列 $a = \langle 1, 2, 4, 8, 16, \dots \rangle$ 的生成函数是 $\sum_{n \geq 0} 2^n x^n$ 。
4. 序列 $a = \langle 1, 3, 5, 7, 9, \dots \rangle$ 的生成函数是 $\sum_{n \geq 0} (2n+1)x^n$ 。

换句话说, 如果序列 a 有通项公式, 那么它的普通生成函数的系数就是通项公式。

基本运算

考虑两个序列 a, b 的普通生成函数, 分别为 $F(x), G(x)$ 。那么有

$$F(x) \pm G(x) = \sum_n (a_n \pm b_n) x^n$$

因此 $F(x) \pm G(x)$ 是序列 $\langle a_n \pm b_n \rangle$ 的普通生成函数。

考虑乘法运算，也就是卷积：

$$F(x)G(x) = \sum_n x^n \sum_{i=0}^n a_i b_{n-i}$$

因此 $F(x)G(x)$ 是序列 $\langle \sum_{i=0}^n a_i b_{n-i} \rangle$ 的普通生成函数。

封闭形式

在运用生成函数的过程中，我们不会一直使用形式幂级数的形式，而会适时地转化为封闭形式以更好地化简。

例如 $\langle 1, 1, 1, \dots \rangle$ 的普通生成函数 $F(x) = \sum_{n \geq 0} x^n$ ，我们可以发现

$$F(x)x + 1 = F(x)$$

那么解这个方程得到

$$F(x) = \frac{1}{1-x}$$

这就是 $\sum_{n \geq 0} x^n$ 的封闭形式。

考虑等比数列 $\langle 1, p, p^2, p^3, p^4, \dots \rangle$ 的生成函数 $F(x) = \sum_{n \geq 0} p^n x^n$ ，有

$$\begin{aligned} F(x)px + 1 &= F(x) \\ F(x) &= \frac{1}{1-px} \end{aligned}$$

等比数列的封闭形式与展开形式是常用的变换手段。

应用

接下来给出一些例题，来介绍生成函数在 OI 中的具体应用。

普通生成函数可以用来解决多重集合组合数问题。

问题：有 n 种物品，每种物品有 a_i 个，问取 m 个物品的组合数？

多重集合组合数

设从每种物品中取 b_i 个， $0 \leq b_i \leq a_i$ ，

$m = \sum_{i=1}^n b_i$ ，对于一组选定的 b_i 进行组合的方案数为 **1**。

例如，取 3 个 A，1 个 B 的方案就是 {AAAB}；取 2 个 A、2 个 B 的方案就是 {AABB}。

那么，所有满足

$b_1 + b_2 + \dots + b_n = m$ 的方案之和，即答案。

构造普通生成函数

第 1 种物品的生成函数为 $(1 + x^1 + x^2 + \dots + x^{a_1})$ ，

第 n 种物品的生成函数为 $(1 + x^1 + x^2 + \dots + x^{a_n})$ 。

即

$$(1 + x^1 + x^2 + \dots + x^{a_1})(1 + x^1 + x^2 + \dots + x^{a_2}) \cdots (1 + x^1 + x^2 + \dots + x^{a_n})$$

求 x^m 的系数。

注意： 指数即物品个数，系数即组合数。

例如：

有三种物品，分别有 3、2、1 个，问取 4 个物品的组合数？

枚举的话，有 {AAAB, AAAC, AABB, AABC, ABBC}，5 个方案。

构造

$$(1 + x + x^2 + x^3)(1 + x + x^2)(1 + x)$$

逐步展开：

$$\begin{aligned} &= (1 + x + x^2 + x^3)(1 + x + x^2)(1 + x) \\ &= (1 + x + x^2 + x^3 + x^2 + x^3 + x^4 + x^3 + x^4 + x^5)(1 + x) \\ &= (1 + 2x + 3x^2 + 3x^3 + 3x^4 + 2x^5 + x^6)(1 + x) \\ &= 1 + 3x + 5x^2 + 6x^3 + 5x^4 + 3x^5 + x^6 \end{aligned}$$

x^4 的系数为 5，即答案。

HDU - 1085 Holding Bin-Laden Captive!

面值为 1, 2, 5 的硬币分别有 a_1, a_2, a_3 枚，
问用这些硬币**不能**组成的最小面值是多少？

思路

构造生成函数：

$$(1 + x^1 + x^2 + \cdots + x^{a_1}) \times (1 + x^2 + x^4 + \cdots + x^{2a_2}) \times (1 + x^5 + x^{10} + \cdots + x^{5a_3})$$

从小到大遍历系数，**为 0 的那一项**就是答案。

例如：

1 分有 1 枚，2 分有 1 枚，5 分有 1 枚：

$$\begin{aligned} &(1 + x^1)(1 + x^2)(1 + x^5) \\ &= (1 + x^2 + x^1 + x^3)(1 + x^5) \\ &= (1 + x^1 + x^2 + x^3)(1 + x^5) \\ &= (1 + x^5 + x^1 + x^6 + x^2 + x^7 + x^3 + x^8) \\ &= 1 + x^1 + x^2 + x^3 + x^5 + x^6 + x^7 + x^8 \end{aligned}$$

最小不能组成的面值是 4。

食物

在许多不同种类的食物中选出 n 个，每种食物的限制如下：

1. 承德汉堡：偶数个
2. 可乐：0 个或 1 个
3. 鸡腿：0 个，1 个或 2 个
4. 蜜桃多：奇数个
5. 鸡块：4 的倍数个
6. 包子：0 个，1 个，2 个或 3 个
7. 土豆片炒肉：不超过一个。
8. 面包：3 的倍数个

每种食物都是以「个」为单位，只要总数加起来是 n 就算一种方案。对于给出的 n 你需要计算出方案数

这是一道经典的生成函数题。对于一种食物，我们可以设 a_n 表示这种食物选 n 个的方案数，并求出它的生成函数。而两种食物一共选 n 个的方案数的生成函数，就是它们生成函数的卷积。多种食物选 n 个的方案数的生成函数也是它们生成函数的卷积。

在理解了方案数可以用卷积表示以后，我们就可以构造生成函数（标号对应题目中食物的标号）：

$$1. \sum_{n \geq 0} x^{2n} = \frac{1}{1 - x^2}.$$

$$2. 1 + x.$$

$$3. 1 + x + x^2 = \frac{1 - x^3}{1 - x}.$$

$$4. \sum_{n \geq 0} x^{2n+1} = \sum_{n \geq 0} x^n - \sum_{n \geq 0} x^{2n} = \frac{x}{1 - x^2}.$$

$$5. \sum_{n \geq 0} x^{4n} = \frac{1}{1 - x^4}.$$

$$6. 1 + x + x^2 + x^3 = \frac{1 - x^4}{1 - x}.$$

$$7. 1 + x.$$

$$8. \sum_{n \geq 0} x^{3n} = \frac{1}{1 - x^3}.$$

那么全部乘起来，得到答案的生成函数：

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

广义二项式定理

$$\frac{1}{(1-x)^n} = \sum_{i=0}^{\infty} C_{n+i-1}^i x^i$$

然后将它转化为展开形式（使用广义二项式定理）：

$$\begin{aligned} F(x) &= x \sum_{i=0}^{\infty} (C_{4+i-1}^i) x^i \\ &= \sum_{i=0}^{\infty} (C_{4+i-1}^i) x^{i+1} \\ &= \sum_{k=1}^{\infty} (C_{k+2}^{k-1}) x^k \\ &= \sum_{k=1}^{\infty} (C_{k+2}^3) x^k \end{aligned}$$

因此答案为

$$C_{n+2}^3$$

指数生成函数

指数生成函数：

$$F(x) = \sum_{n \geq 0} a_n \frac{x^n}{n!}$$

序列 $\langle 1, 1, 1, \dots \rangle$ 的指数生成函数是

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{n \geq 0} \frac{x^n}{n!} = e^x$$

序列 $\langle 1, p, p^2, \dots \rangle$ 的指数生成函数是

$$1 + p \frac{x}{1!} + p^2 \frac{x^2}{2!} + p^3 \frac{x^3}{3!} + \dots = \sum_{n \geq 0} p^n \frac{x^n}{n!} = e^{px}$$

基本运算

加减运算

$$F(x) \pm G(x) = \sum_{i \geq 0} a_i \frac{x^i}{i!} \pm \sum_{j \geq 0} b_j \frac{x^j}{j!} = \sum_{n \geq 0} (a_n \pm b_n) \frac{x^n}{n!}$$

因此 $F(x) \pm G(x)$ 是序列 $\langle a_n \pm b_n \rangle$ 的指数生成函数。

乘法运算（卷积）

$$F(x)G(x) = \sum_{i \geq 0} a_i \frac{x^i}{i!} \sum_{j \geq 0} b_j \frac{x^j}{j!} = \sum_{n \geq 0} x^n \sum_{i=0}^n a_i b_{n-i} \frac{1}{i!(n-i)!} = \sum_{n \geq 0} \frac{x^n}{n!} \sum_{i=0}^n \frac{n!}{i!(n-i)!} a_i b_{n-i} =$$

因此 $F(x)G(x)$ 是序列 $\langle \sum_{i=0}^n C_n^i a_i b_{n-i} \rangle$ 的指数生成函数。

封闭形式

我们同样考虑指数生成函数的封闭形式。

序列 $\langle 1, 1, 1, \dots \rangle$ 的指数生成函数是：

$$\hat{F}(x) = \sum_{n \geq 0} \frac{x^n}{n!} = e^x$$

因为你将 e^x 在 $x = 0$ 处泰勒展开就得到了它的无穷级数形式。

类似地，等比数列 $\langle 1, p, p^2, \dots \rangle$ 的指数生成函数是：

$$\hat{F}(x) = \sum_{n \geq 0} \frac{p^n x^n}{n!} = e^{px}$$

指数生成函数可以用来解决多重集排列数问题。

HDU - 1521 排列组合

题意：有 n 种物品，每种物品有 a_i 个，问取 m 个物品的排列数？

多重集排列数

设从每种物品中取 b_i 个， $0 \leq b_i \leq a_i$ ， $m = \sum_{i=1}^n b_i$ ，对于一组选定的 b_i 进行排列的方案数为 $\frac{m!}{b_1!b_2!\cdots b_n!}$ 。若 m 个物品互不相同，其排列数为 $m!$ ，分母就是对每种相同物品的排列数去重。

例如，取3个A、1个B的排列数为 $\frac{4!}{3!1!} = \frac{24}{6} = 4$ ，即 {AAAA, AABA, ABAA, BAAA}。

取2个A、2个B的排列数为 $\frac{4!}{2!2!} = \frac{24}{4} = 6$ ，即 {AABB, ABAB, ABBA, BAAB, BABA, BBAA}。

那么，所有满足 $b_1 + b_2 + \cdots + b_n = m$ 的排列数之和，即答案。

构造指数生成函数

第1种物品的生成函数为 $(1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^{a_1}}{a_1!})$ ，第 n 种物品的生成函数为 $(1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^{a_n}}{a_n!})$ 。

即 $(1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^{a_1}}{a_1!})(1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^{a_2}}{a_2!}) \cdots (1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^{a_n}}{a_n!})$ ，求 $\frac{x^m}{m!}$ 的系数。

做乘法， $\frac{x^{b_1}}{b_1!} \times \frac{x^{b_2}}{b_2!} \times \cdots \times \frac{x^{b_n}}{b_n!} = \frac{x^{b_1+b_2+\cdots+b_n}}{b_1!b_2!\cdots b_n!} = \frac{x^m}{b_1!b_2!\cdots b_n!} = \frac{m!}{b_1!b_2!\cdots b_n!} \cdot \frac{x^m}{m!}$ 。

做卷积，所有满足 $b_1 + b_2 + \cdots + b_n = m$ 的项的系数之和，再乘以 $m!$ ，即答案。

一点小结论（前已述及）

(1) 序列 a 的普通生成函数： $F(x) = \sum a_n x^n$

(2) 序列 a 的指数生成函数： $F(x) = \sum a_n \frac{x^n}{n!}$

泰勒展开式

普通生成函数：

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots = \sum_{n=0}^{\infty} x^n$$

$$\frac{1}{1-x^2} = 1 + x^2 + x^4 + \dots$$

$$\frac{1}{1-x^3} = 1 + x^3 + x^6 + \dots$$

$$\frac{1}{(1-x)^2} = 1 + 2x + 3x^2 + \dots$$

指数生成函数：

$$e^x = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

$$e^{-x} = 1 - \frac{x^1}{1!} + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots$$

$$\frac{e^x + e^{-x}}{2} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots$$

$$\frac{e^x - e^{-x}}{2} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots$$

有穷序列的生成函数

$$1 + x + x^2 = \frac{1 - x^3}{1 - x}$$

$$1 + x + x^2 + x^3 = \frac{1 - x^4}{1 - x}$$

广义二项式定理

$$\frac{1}{(1 - x)^n} = \sum_{i=0}^{\infty} C_{n+i-1}^i x^i$$

证明:

二项式定理:

$$(1 + x)^n = \sum_{i=0}^n C_n^i x^i$$

(1) 扩展域:

$$(1 + x)^n = \sum_{i=0}^{\infty} C_n^i x^i, \quad \text{当 } i > n \text{ 时 } C_n^i = 0$$

(2) 扩展指数为负数:

$$\begin{aligned} C_{-n}^i &= (-n)(-n-1)\cdots(-n-i+1) \\ &= (-1)^i \cdot \frac{n(n+1)\cdots(n+i-1)}{i!} = (-1)^i C_{n+i-1}^i \end{aligned}$$

$$(1 + x)^{-n} = \sum_{i=0}^{\infty} C_{-n}^i x^i$$

$$= \sum_{i=0}^{\infty} (-1)^i C_{n+i-1}^i x^i$$

(3) 括号内的加号变减号：

$$\begin{aligned}(1-x)^{-n} &= \sum_{i=0}^{\infty} (-1)^i C_{n+i-1}^i (-x)^i \\ &= \sum_{i=0}^{\infty} C_{n+i-1}^i x^i\end{aligned}$$

证毕。

莫反

狄利克雷生成函数

数列 $\langle a_1, a_2, a_3, \dots \rangle$ 的狄利克雷生成函数定义为：

$$F(x) = \frac{a_1}{1^x} + \frac{a_2}{2^x} + \frac{a_3}{3^x} + \dots = \sum_{n=1}^{\infty} \frac{a_n}{n^x}$$

乘法运算（Dirichlet 卷积）

$$\begin{aligned}\sum_{i=1}^{\infty} \frac{a_i}{i^x} \sum_{j=1}^{\infty} \frac{b_j}{j^x} &= \left(\frac{a_1}{1^x} + \frac{a_2}{2^x} + \frac{a_3}{3^x} + \frac{a_4}{4^x} + \dots \right) \left(\frac{b_1}{1^x} + \frac{b_2}{2^x} + \frac{b_3}{3^x} + \frac{b_4}{4^x} + \dots \right) \\ &= \frac{a_1 b_1}{1^x} + \frac{a_1 b_2}{2^x} + \frac{a_2 b_1}{3^x} + \frac{a_1 b_3}{4^x} + \dots \\ &= \sum_{n=1}^{\infty} \frac{1}{n^x} \sum_{d|n} a_d b_{\frac{n}{d}}\end{aligned}$$

系数计算规则

$\frac{1}{n^x}$ 项的系数等于所有满足 $d|n$ (d 整除 n) 的项 $a_d b_{n/d}$ 之和：

- 4^x 的系数: $a_1b_4 + a_2b_2 + a_4b_1$
(枚举 4 的约数 $d = 1, 2, 4$)
- 6^x 的系数: $a_1b_6 + a_2b_3 + a_3b_2 + a_6b_1$
(枚举 6 的约数 $d = 1, 2, 3, 6$)

一点和式的小结论

欧拉函数

1. 定义

欧拉函数 $\varphi(n)$ 表示小于等于 n 且与 n 互质的正整数个数:

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

欧拉函数值表

n	1	2	3	4	5	6	7	8	9	10	11	12
$\varphi(n)$	1	1	2	2	4	2	6	4	6	4	10	4

2. 性质

欧拉函数求和定理

对于任意正整数 n , 其所有因子的欧拉函数值之和等于 n :

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

验证示例

$$\begin{aligned}\varphi(1) &= 1 \\ \varphi(1) + \varphi(2) &= 1 + 1 = 2 \\ \varphi(1) + \varphi(3) &= 1 + 2 = 3 \\ \varphi(1) + \varphi(2) + \varphi(4) &= 1 + 1 + 2 = 4 \\ \varphi(1) + \varphi(5) &= 1 + 4 = 5 \\ \varphi(1) + \varphi(2) + \varphi(3) + \varphi(6) &= 1 + 1 + 2 + 2 = 6\end{aligned}$$

证明思路

考虑以 n 为分母的真分数 $[0, 1)$ 区间：

$$\frac{0}{n}, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n-1}{n}$$

将这些分数化简为最简形式后，根据分母分组，可证明结论。

示例演示 ($n = 12$)

所有真分数

$$\frac{0}{12}, \frac{1}{12}, \frac{2}{12}, \frac{3}{12}, \frac{4}{12}, \frac{5}{12}, \frac{6}{12}, \frac{7}{12}, \frac{8}{12}, \frac{9}{12}, \frac{10}{12}, \frac{11}{12}$$

化简后分组

分母 d	最简分数	个数 $\varphi(d)$
1	$\frac{0}{1}$	$\varphi(1) = 1$
2	$\frac{1}{2}$	$\varphi(2) = 1$
3	$\frac{1}{3}, \frac{2}{3}$	$\varphi(3) = 2$

分母 d	最简分数	个数 $\varphi(d)$
4	$\frac{1}{4}, \frac{3}{4}$	$\varphi(4) = 2$
6	$\frac{1}{6}, \frac{5}{6}$	$\varphi(6) = 2$
12	$\frac{1}{12}, \frac{5}{12}, \frac{7}{12}, \frac{11}{12}$	$\varphi(12) = 4$

验证等式

$$\sum_{d|12} \varphi(d) = \varphi(1) + \varphi(2) + \varphi(3) + \varphi(4) + \varphi(6) + \varphi(12) = 1 + 1 + 2 + 2 + 2 + 4 = 12$$

注意到窝们的证明思路天然满足定义

一般性证明

1. 考虑所有分母为 n 的真分数：

$$\frac{k}{n} \quad (0 \leq k < n)$$

共有 n 个分数。

2. 将每个分数化简为最简形式 $\frac{a}{d}$, 其中 $d \mid n$ 且 $\gcd(a, d) = 1$ 。

3. 对 n 的每个约数 d 分组：

- 分母为 d 的分数个数 = $\varphi(d)$
- 因为分子 a 需满足 $1 \leq a \leq d$ 且 $\gcd(a, d) = 1$

4. 总和为：

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

即得证。

莫比乌斯函数

1. 定义

莫比乌斯函数 $\mu(n)$ 定义如下：

$$\mu(n) = \begin{cases} 1 & \text{若 } n = 1 \\ (-1)^s & \text{若 } n = p_1 p_2 \cdots p_s \text{ (无平方因子的整数)} \\ 0 & \text{若 } n \text{ 包含平方因子} \end{cases}$$

莫比乌斯函数值表

n	1	2	3	4	5	6	7	8	9	10	11	12
$\mu(n)$	1	-1	-1	0	-1	1	-1	0	0	1	-1	0

2. 核心性质

$$\sum_{d|n} \mu(d) = [n = 1] = \begin{cases} 1 & n = 1 \\ 0 & n > 1 \end{cases}$$

验证示例

$$n = 1: \mu(1) = 1$$

$$n = 2: \mu(1) + \mu(2) = 1 + (-1) = 0$$

$$n = 3: \mu(1) + \mu(3) = 1 + (-1) = 0$$

$$n = 4: \mu(1) + \mu(2) + \mu(4) = 1 + (-1) + 0 = 0$$

$$n = 6: \mu(1) + \mu(2) + \mu(3) + \mu(6) = 1 + (-1) + (-1) + 1 = 0$$

3. 证明 ($n > 1$ 时和为 0)

证明思路

设 $n = p_1^{a_1} p_2^{a_2} \cdots p_s^{a_s}$, 定义 n' 为 n 的平方自由部分:

$$n' = p_1 p_2 \cdots p_s$$

则:

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

组合证明

考虑 n 的质因子集合 S , 其大小为 s :

- $\mu(d) \neq 0$ 的 d 对应 S 的子集
- d 的质因子个数为 k 时, $\mu(d) = (-1)^k$
- 由二项式定理

$$\sum_{d|n'} \mu(d) = \sum_{k=0}^s (-1)^k \binom{s}{k} = (1 + (-1))^s = 0$$

示例说明 ($n = 6$)

$6 = 2^1 \times 3^1$, $S = \{2, 3\}$:

$\mu(1) = (-1)^0 = 1$	(取 0 个质因子)
$\mu(2) = (-1)^1 = -1$	(取质因子 2)
$\mu(3) = (-1)^1 = -1$	(取质因子 3)
$\mu(6) = (-1)^2 = 1$	(取质因子 2, 3)

和为 $1 + (-1) + (-1) + 1 = 0$ 。

狄利克雷卷积

定义

设 $f(n), g(n)$ 是两个积性函数，其狄利克雷卷积定义为：

$$(f * g)(n) = \sum_{d|n} f(d)g\left(\frac{n}{d}\right) = \sum_{d|n} f\left(\frac{n}{d}\right)g(d)$$

注意跟狄利克雷生成函数形式上的相似性

读作： f 卷 g

示例

$$(f * g)(4) = f(1)g(4) + f(2)g(2) + f(4)g(1)$$

运算规律

- 交换律**: $f * g = g * f$
- 结合律**: $(f * g) * h = f * (g * h)$
- 分配律**: $(f + g) * h = f * h + g * h$

常用函数

函数名称	符号表示	定义
元函数	$\epsilon(n)$	$[n = 1] = \begin{cases} 1 & n = 1 \\ 0 & n > 1 \end{cases}$
常数函数	$1(n)$	1
恒等函数	$id(n)$	n

函数名称	符号表示	定义
欧拉函数	$\varphi(n)$	$<n$ 且与 n 互质的数的个数
莫比乌斯函数	$\mu(n)$	$\begin{cases} 1 & n = 1 \\ (-1)^k & n = p_1 p_2 \dots p_k \\ 0 & n \text{ 含平方因子} \end{cases}$

注意符号的读法 μ 读作“缪”， φ 读作“phi”， ϵ 读作“一穆西隆”

常用卷积关系

简记形式

- $\sum_{d|n} \mu(d) = [n = 1] \Leftrightarrow \mu * 1 = \epsilon$
- $\sum_{d|n} \varphi(d) = n \Leftrightarrow \varphi * 1 = id$
- $\sum_{d|n} \mu(d) \frac{n}{d} = \varphi(n) \Leftrightarrow \mu * id = \varphi$
- $f * \epsilon = f$
- $f * 1 \neq f$

注意莫比乌斯函数是常数函数的逆元

证明

1. $\mu * 1 = \epsilon$

$$(\mu * 1)(n) = \sum_{d|n} \mu(d) \cdot 1\left(\frac{n}{d}\right) = \sum_{d|n} \mu(d) = [n = 1] = \epsilon(n)$$

2. $\varphi * 1 = id$

$$(\varphi * 1)(n) = \sum_{d|n} \varphi(d) \cdot 1\left(\frac{n}{d}\right) = \sum_{d|n} \varphi(d) = n = id(n)$$

$$3. \mu * id = \varphi$$

$$(\mu * id)(n) = \sum_{d|n} \mu(d) \cdot id\left(\frac{n}{d}\right) = \sum_{d|n} \mu(d) \cdot \frac{n}{d} = \varphi(n)$$

$$\mu * id = \mu * \varphi * 1 = (\varphi * 1 * \mu) = \epsilon * \mu = \varphi$$

$$4. f * \epsilon = f$$

$$(f * \epsilon)(n) = \sum_{d|n} f(d) \cdot \epsilon\left(\frac{n}{d}\right) = \sum_{d|n} f(d) \cdot \left[\frac{n}{d} = 1\right] = f(n)$$

$$5. f * 1 \neq f$$

$$(f * 1)(n) = \sum_{d|n} f(d) \cdot 1\left(\frac{n}{d}\right) = \sum_{d|n} f(d) \neq f(n)$$

莫比乌斯反演

其实就是一下几个式子(条件式变成和式)

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

$$\sum_{d|n} \mu(d) = [n = 1] = \begin{cases} 1 & n = 1 \\ 0 & n > 1 \end{cases}$$

把 n 换成 $\gcd(a, b)$

$$\sum_{d|\gcd(a,b)} \mu(d) = [\gcd(a, b) = 1] \begin{cases} 1 & \gcd(a, b) = 1 \\ 0 & \gcd(a, b) > 1 \end{cases}$$

莫比乌斯变换

基本公式

设 $f(n), g(n)$ 均为积性函数，则：

$$f(n) = \sum_{d|n} g(d) \iff g(n) = \sum_{d|n} \mu(d) f\left(\frac{n}{d}\right)$$

即

$$f = g * 1 \iff g = \mu * f$$

- $f(n)$ 称为 $g(n)$ 的莫比乌斯变换
- $g(n)$ 称为 $f(n)$ 的莫比乌斯逆变换

注意对于一些函数 $f(n)$ ，如果很难直接求出它的值，而容易求出其倍数和或约数和 $g(n)$ ，那么可以通过莫比乌斯反演简化运算，求得 $f(n)$ 的值。

证明方法一（卷积形式）

正向推导

$$\begin{aligned} f &= g * 1 \\ \mu * f &= \mu * (g * 1) \\ &= g * (\mu * 1) \\ &= g * \epsilon \\ &= g \end{aligned}$$

逆向推导

$$\begin{aligned}
g &= \mu * f \\
g * 1 &= (\mu * f) * 1 \\
&= f * (\mu * 1) \\
&= f * \epsilon \\
&= f
\end{aligned}$$

证明方法二（双重求和）

$$\begin{aligned}
\sum_{d|n} \mu(d) f\left(\frac{n}{d}\right) &= \sum_{d|n} \mu(d) \sum_{k|\frac{n}{d}} g(k) \\
&= \sum_{d|n} \sum_{k|\frac{n}{d}} \mu(d) g(k) \\
&= \sum_{k|n} \sum_{d|\frac{n}{k}} \mu(d) g(k) \\
&= \sum_{k|n} g(k) \left(\sum_{d|\frac{n}{k}} \mu(d) \right) \\
&= \sum_{k|n} g(k) \cdot \epsilon\left(\frac{n}{k}\right) \\
&= g(n)
\end{aligned}$$

排列组合

圆排列

n 个不同元素围成一圈的**圆排列数**，记作 Q_n^n 。

考虑其中已经排好的一圈，从不同位置断开，会变成 n 个不同的线排列：

$$Q_n^n \times n = A_n^n$$

则

$$Q_n^n = \frac{A_n^n}{n} = (n-1)!$$

例如, 3 个不同元素的圆排列数为 $(3-1)! = 2$ 种:

从 n 个不同元素中选 m 个围成一圈的**圆排列数**, 记作 Q_n^m :

$$Q_n^m = C_n^m \cdot Q_m^m = \frac{n!}{m \cdot (n-m)!}$$

其实就是全排列固定了一个数

错位排列

错位排列是没有任何元素出现在其有序位置的排列。对于 $1 \sim n$ 的排列 P , 如果满足 $P_i \neq i$, 则称 P 是 n 的错位排列。

错位排列数

- $D_1 = 0$
- $D_2 = 1$ (即 $\{2, 1\}$)
- $D_3 = 2$ (即 $\{2, 3, 1\}, \{3, 1, 2\}$)
- $D_4 = 9$ (即 $\{2, 1, 4, 3\}, \{2, 3, 4, 1\}, \{2, 4, 1, 3\}, \{3, 1, 4, 2\}, \{3, 4, 1, 2\}, \{3, 4, 2, 1\}, \{4, 1, 2, 3\}, \{4, 1, 3, 2\}$)

递推关系 $D_n = (n-1)(D_{n-1} + D_{n-2})$

边界条件:

$$D_1 = 0, \quad D_2 = 1$$

n	1	2	3	4	5	6	7	8
D_n	0	1	2	9	44	265	1854	14833

信封问题

n 封不同的信（编号 $1, 2, \dots, n$ ）放入 n 个编号对应的信封中，要求每个信封的编号与信的编号都不相同。有多少种放置方法？

递推关系分析

考虑第 n 封信的放置：

- 情况1：**前 $n - 1$ 封信已全错排
 - 第 n 封信只需与前面任一封信交换位置
 - 方法数： $(n - 1) \cdot D_{n-1}$
- 情况2：**前 $n - 1$ 封信恰好有 1 封位置正确
 - 第 n 封信必须与位置正确的信交换
 - 方法数： $(n - 1) \cdot D_{n-2}$

其他情况无法通过一次交换变成全错排

第一类斯特林数（斯特林轮换数）

将 n 个不同元素划分为 m 个非空圆排列的方案数，
记作 $S(n, m)$ 或：

$$\begin{bmatrix} n \\ m \end{bmatrix}$$

递推关系

$$\begin{bmatrix} n \\ m \end{bmatrix} = \begin{bmatrix} n - 1 \\ m - 1 \end{bmatrix} + (n - 1) \begin{bmatrix} n - 1 \\ m \end{bmatrix}$$

组合解释（圆桌问题）

n 个人坐 m 张圆桌的方案数，考虑第 n 个人的两种坐法：

1. 单独坐一桌

- 前 $n - 1$ 人坐满剩余的 $m - 1$ 张桌
- 方案数： $\left[\begin{matrix} n - 1 \\ m - 1 \end{matrix} \right]$

2. 与其他人同坐

- 前 $n - 1$ 人先坐满 m 张桌
- 第 n 人可坐到任意 $n - 1$ 个人的左侧
- 方案数： $(n - 1) \cdot \left[\begin{matrix} n - 1 \\ m \end{matrix} \right]$

第二类斯特林数（斯特林子集数）

将 n 个不同元素划分为 m 个非空子集的方案数，
记作 $S(n, m)$ 或：

$$\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$$

递推关系

$$\left\{ \begin{matrix} n \\ m \end{matrix} \right\} = \left\{ \begin{matrix} n - 1 \\ m - 1 \end{matrix} \right\} + m \left\{ \begin{matrix} n - 1 \\ m \end{matrix} \right\}$$

组合解释（房间分配问题）

n 个人进入 m 个房间的方案数（每个房间非空），考虑第 n 个人的两种选择：

1. 单独进入新房间

- 前 $n - 1$ 人进入剩余的 $m - 1$ 个房间
- 方案数： $\left\{ \begin{matrix} n - 1 \\ m - 1 \end{matrix} \right\}$

2. 进入已有人的房间

- 前 $n - 1$ 人先进入所有 m 个房间
- 第 n 人可选择进入任意一个已有人的房间
- 方案数: $m \cdot \left\{ \begin{matrix} n - 1 \\ m \end{matrix} \right\}$

Catalan数 通项公式

$$(1) H_n = C_{2n}^n - C_{2n}^{n-1}$$

$$(2) H_n = \frac{1}{n+1} C_{2n}^n$$

$$(3) H_n = \frac{4n-2}{n+1} H_{n-1}$$

证明

卡特兰数 (Catalan)

以走网格为例，从格点 $(0,0)$ 走到格点 (n,n) ，只能向右或向上走，并且不能越过对角线的路径的条数，就是卡特兰数，记为 H_n 。

通项公式

$$(1) H_n = C_{2n}^n - C_{2n}^{n-1} \quad (2) H_n = \frac{1}{n+1} C_{2n}^n \quad (3) H_n = \frac{4n-2}{n+1} H_{n-1}$$

证明(1)式

先求路径总数，在 $2n$ 次移动中选 n 次向右移动，即 C_{2n}^n 。

再求非法路径，即越过对角线的路径。

把 $y = x + 1$ 这条线画出来，碰到即说明是一条非法路径。

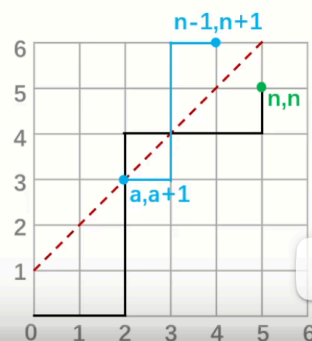
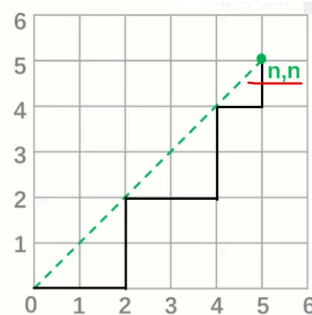
所有的非法路径与这条线有至少一个交点，把第一个交点设为 $(a, a+1)$ ，把 $(a, a+1)$ 之后的路径全部按照 $y = x + 1$ 这条线对称过去，这样，最后的终点就会变成 $(n-1, n+1)$ 。

所有非法路径对称后都唯一对应着一条到 $(n-1, n+1)$ 的路径，所以非法路径数就是 C_{2n}^{n-1} ，合法路径数就是 $C_{2n}^n - C_{2n}^{n-1}$ 。

证明(2)式

$$H_n = C_{2n}^n - C_{2n}^{n-1} = \frac{(2n)!}{n!n!} - \frac{(2n)!}{(n+1)!(n-1)!}$$

$$= \frac{(2n)!}{n!(n-1)!} \left(\frac{1}{n} - \frac{1}{n+1} \right) = \frac{(2n)!}{n!n!(n+1)} = \frac{1}{n+1} C_{2n}^n$$



Catalan 数列

H_0	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8
1	1	2	5	14	42	132	429	1430

Catalan 特征

从 $(0,0)$ 到 (n,n) ，不越过对角线，即任何时候，向上走的步数不能超过向右走的步数。

一种操作数不能超过另外一种操作数，或者两种操作不能有交集，这些操作的合法方案数，通常是卡特兰数。

Catalan 应用

1. 一个有 n 个 0 和 n 个 1 组成的字串，且所有的前缀字串皆满足 1 的个数不超过 0 的个数。这样的字串个数有多少？
2. 包含 n 组括号的合法运算式的个数有多少？
3. 一个栈的进栈序列为 $1, 2, 3, \dots, n$ ，有多少个不同的出栈序列？
合法性：任何时刻不能空栈出栈 \Rightarrow 任意前缀“)”不多于“(”；最后入栈、出栈各 n 次，栈空。
4. n 个结点可构造多少个不同的二叉树？
5. 在圆上选择 $2n$ 个点，将这些点成对连接起来使得所得的 n 条弦不相交的方法数？
6. 通过连结顶点而将 $n + 2$ 边的凸多边形分成 n 个三角形的方法数？

说明

这些都是卡特兰数，因为它们都与**合法括号序列** / **Dyck 路径**存在天然的双射，或者都满足同一个**卡特兰递推**

$$C_0 = 1, \quad C_n = \sum_{i=0}^{n-1} C_i C_{n-1-i} \quad (n \geq 1),$$

并且初值一致，所以计数相同。分别说——

3) 单栈出栈序列的个数 = C_n

把栈的操作写成长度 $2n$ 的串：

- 入栈记“(”，出栈记”)”。

合法性：任何时刻不能空栈出栈 \Rightarrow 任意前缀“)”不多于“(”；最后入栈、出栈各 n

次。

这正是**合法括号序列**的定义，因此个数为 $C_n = \frac{1}{n+1} \binom{2n}{n}$ 。

(同说法：栈可生成的排列=231-避免排列，其数目为卡特兰数。)

4) n 个结点的二叉树个数 = C_n

设 T_n 为有 n 个结点（有序、无标号）的二叉树数。以根为界：左子树 i 个结点、右子树 $n - 1 - i$ 个结点，二者独立：

$$T_n = \sum_{i=0}^{n-1} T_i T_{n-1-i}, \quad T_0 = 1.$$

这恰是卡特兰递推，因此 $T_n = C_n$ 。

（等价双射：对二叉树做先序/中序边走访，“向下”记“(”，“返回”记“)”，得到 Dyck 串，反之亦然。）

5) 圆上 $2n$ 点配对且弦不相交的配法数 = C_n

固定点 1，它必须与某个点 $2k$ 相连（顺时针计）。这一条弦把圆分成两侧：

- 一侧有 $2k - 2$ 个点，可不交配对数 C_{k-1} ；
- 另一侧有 $2n - 2k$ 个点，可不交配对数 C_{n-k} 。

枚举 $k = 1..n$ 得

$$M_n = \sum_{k=1}^n C_{k-1} C_{n-k} = \sum_{i=0}^{n-1} C_i C_{n-1-i}.$$

同初值 $M_0 = 1$ ，故 $M_n = C_n$ 。

6) 将凸 $(n + 2)$ -边形三角剖分的方法数 $= C_n$

固定顶点 1, 选一条对角线 $(1, j)$ ($j = 3..n + 2$)。它把多边形分成:

- 一个 $(j - 1)$ -边形 (可三角剖分数 C_{j-3}) ;
- 一个 $(n + 3 - j)$ -边形 (可三角剖分数 $C_{n-(j-2)}$) 。

求和得同一递推:

$$T_n = \sum_{j=3}^{n+2} C_{j-3} C_{n-(j-2)} = \sum_{i=0}^{n-1} C_i C_{n-1-i}, \quad T_0 = 1,$$

故 $T_n = C_n$ 。

递推来源

从合法括号序列 (Dyck 路径) 推导

考虑长度 $2n$ 的合法括号序列:

- 第一个符号必然是“(”。
- 找到与之匹配的”)”位置。假设它在第 $2k + 2$ 个位置 (即括号包住了 k 对括号) 。

于是序列可以分成三部分:

$$\underbrace{(\quad S \quad)}_{2k \text{ 长度的合法串}} \quad \underbrace{T}_{2(n-k-1) \text{ 长度的合法串}}$$

- 内部 S 是一个合法串, 长度 $2k$, 个数 C_k ;
- 外部 T 也是一个合法串, 长度 $2(n - k - 1)$, 个数 C_{n-1-k} 。

所以

$$C_n = \sum_{k=0}^{n-1} C_k \cdot C_{n-1-k}.$$

这就是递推的来源。

勒让德公式

勒让德公式是什么

勒让德公式用来算质数 p 在阶乘 $n!$ 的质因子分解中的指数。

公式是：

$$v_p(n!) = \sum_{k=1}^{\infty} \left\lfloor \frac{n}{p^k} \right\rfloor,$$

其中 $v_p(n!)$ 表示 p 在 $n!$ 里出现的次数（指数）。

为什么成立

考虑 $n! = 1 \cdot 2 \cdot 3 \cdots n$ 。

- 每隔 p 个数里有一个能被 p 整除，所以至少有 $\lfloor n/p \rfloor$ 个因子 p ；
 - 每隔 p^2 个数里有一个能被 p^2 整除，它会额外贡献一个 p ，所以再加 $\lfloor n/p^2 \rfloor$ ；
 - 每隔 p^3 个数里有一个能被 p^3 整除，它会再额外贡献一个 p ……
- 如此类推，直到 $p^k > n$ 为止。

所以总和就是上面的式子。

一个例子

算 $v_2(10!)$ ：

$$\lfloor 10/2 \rfloor + \lfloor 10/4 \rfloor + \lfloor 10/8 \rfloor = 5 + 2 + 1 = 8.$$

验证：

$$10! = 3628800 = 2^8 \cdot 3^4 \cdot 5^2 \cdot 7.$$

在这题里的作用

我们要计算：

$$C_n = \frac{1}{n+1} \binom{2n}{n}.$$

其中涉及到阶乘：

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2}.$$

为了在**任意模数** p 下算出结果（尤其是 p 不是质数时），不能用逆元，要直接做**质因子分解**。于是对每个质数 q ，用勒让德公式求它在 $(2n)!$ 、 $n!$ 中的指数差，得到 q 在组合数中的幂次。

再减去 $n+1$ 的质因子指数，就是卡特兰数的质因子分解。最后用快速幂拼起来，就得到答案 $\text{mod } p$ 。

以下为记忆化搜索函数 `dfs` 的常设定的形参

- **pos**: `int` 型变量，表示当前枚举的位置，一般从高到低
- **limit**: `bool` 型变量，表示枚举的第 `pos` 位是否受到限制
 - 为 `true` 表示取的数不能大于 `a[pos]`，而只有在 `[pos+1, len]` 的位置上填写的数都等于 `a[]` 时该位才为 `true`
 - 否则表示当前位没有限制，可以取到 `[0, R-1]`，因为 `R` 进制的数中数位最多能取到的就是 `R-1`
- **last**: `int` 型变量，表示上一位（第 `pos+1` 位）填写的值
 - 往往用于约束相邻数位之间的关系的问题
- **lead0**: `bool` 型变量，表示是否有前导零，即在 `len → (pos+1)` 这些位置是不是都是前导零
 - 基于常识，我们往往默认一个数没有前导零，也就是最高位不能为 0，即不会写为 `000123`，而是写为 `123`
 - 只有没有前导零的时候，才能计算贡献
 - 那么前导零可用时答案有关？

- 统计 0 出现次数
 - 相邻数位的差值
 - 以后问题为起点确定的奇偶位
- **sum:** int 型变量, 表示当前 $len \rightarrow (pos+1)$ 的数位和
- **r:** int 型变量, 表示整数 x 取模某个数 m 的余数
 - 该参数一般会固定在: 约束中出现了“能被 m 整除”
 - 当然也可以拓展成取数权取模的结果
- **st:** int 型变量, 用于状态压缩
 - 对一个集合的数位权上的出现次数的奇偶性有要求时, 其二进制形式就可以表示每个数出现的奇偶性