概述

信息学的数学板块内容

数论
离散数学
组合数学、组合计数
计算几何*
群论*
…

记住结论的基础上，理解并学会其证明
当然，与数竞不同，大多情况下会用结论就行

概述

数学知识在OI中是一个很重要的部分
例如2021年的NOIP(原提高-省选阶段)演变成了“数学竞赛”
高阶段的信息学竞赛，除了算法应用，数学功底也是考查的一方面

总而言之，多学点数学对信息学也有好处
今天的内容概念居多，证明提供但不怎么会详细讲，然后在题单里实践

信息学

基础数论一

西南大学附属中学校

信息奥赛教练组

一些常见的数学运算符号

求和(累加)

$$\sum$$

$$\sum_{i=1}^{n} a_i$$

```
for(i=1;i<=n;i++) s=s+a[i]
```

求积(累乘)

$$\prod$$

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

```
for(i=1;i<=n;i++) s=s*a[i]
```

约数

模(mod)运算与整除

取模

取得余数%

a对b取模得到的结果就是a除以b的余数

记作a mod b，a%b

整除

a%b模数为0，记作b|a，a是b的倍数

模的性质

模的几个基本性质:

- (a + b)%p = (a%p + b%p)%p
- (a - b)%p = (a%p – b%p)%p
- (a - b)%p = (a - b + p)%p
- a*b%p = (a%p)*(b%p) %p

大家之前做题的时候已经在应用这些性质了
计算中取模，可以避免中间结果溢出
负数的模数如果没有特殊约定，则为最小的正整数
负数取模的方法：不断加模数，直到为正，即为余数

剩余系与缩系(了解)

"剩余系"，就是指对于某一个特定的正整数n，一个整数集中的数模n所得的余数域

一个剩余系中包含了这个正整数所有可能的余数（一般地，对于任意正整数n，有n个余数：0,1,2,...,n-1），那么就被称为是模n的一个完全剩余系

简化剩余系也称既约剩余系或缩系，是m的完全剩余系中与m互素的数构成的子集，如果模m的一个剩余类里所有数都与m互素，就把它叫做与模m互素的剩余类。
在与模m互素的全体剩余类中，从每一个类中各任取一个数作为代表组成的集合，叫做模m的一个简化剩余系。
例如，模5的一个简化剩余系是1，2，3，4，模10的一个简化剩余系是1，3，7，9，模18的一个简化剩余系是1，5，7，11，13，17。

gcd与lcm

GCD

如果a%x = 0，我们称x是a的约数，也称a是x的倍数
a与b的最大公约数，是指一个最大的整数x，使得x同时是a和b的约数
a与b的最大公约数记作gcd(a,b)

LCM

两个或多个整数公有的倍数叫做它们的公倍数
其中除0以外最小的一个公倍数就叫做这几个整数的最小公倍数。
a与b的最小公倍数记作lcm(a,b)

本质上求解gcd和lcm是一样的，都需要求解gcd

如何求解gcd?

欧几里得算法(辗转相除法)

欧几里得算法又称辗转相除法
算法公式：

$$\gcd(a,b)=\begin{cases}\gcd(b,a\%b), & b\neq 0\\ a, & b=0\end{cases}$$

时间复杂度为log级别

```
int gcd(int a,int b)
{    if(b==0)return a;
         else return gcd(b,a%b);
}
```

欧几里得算法

证明：gcd(a,b)=gcd(b,a%b)

证明过程：
设gcd(a,b)=d，a=md，b=nd
则gcd(m,n)=1(也称m与n互质)

$$a\%b = a - \left\lfloor \frac{a}{b} \right\rfloor * b = md - \left\lfloor \frac{md}{nd} \right\rfloor * nd$$
$$= \left(m - \left\lfloor \frac{m}{n} \right\rfloor * n\right) d$$
$$= m\%n * d$$

所以gcd(b,a%b)=gcd(nd,m%n*d)=d*gcd(n,m%n)
因此只要证gcd(n,m%n)=1

欧几里得算法

证明：**gcd(n,m%n)=1**

证明过程:

假设gcd(n,m%n)=q≠1

设m=kn+r (0<=r<n)，则m%n=r，gcd(n,r)=q

设n=n’q，r=r’q

那么m=kn’q+r’q=(kn’+r’)q

那么q就成为m与n的公约数，这与我们所假设的gcd(m,n)=1矛盾(反证法)

所以gcd(n,m%n)=1

所以gcd(b,a%b) = d*gcd(n,m%n) = d = gcd(a,b)

证明了该算法

gcd扩展性质:

gcd(a,b,c)=gcd(gcd(a,b),c)

将a与b的最小公倍数记作lcm(a,b)

则lcm(a,b)=a*b/gcd(a,b)

(证明用质数的唯一分解定理证)

与gcd同理，lcm(a,b,c)=lcm(lcm(a,b),c)

但是lcm(a,b,c)=a*b*c/gcd(a,b,c)不成立

质数

质(素)数

这个大家小学应该学过……

质数，又称素数，是指除1和本身外没有其他约数的正整数，

例如2,3,5,7,11……

否则称为合数

质数在数论中十分常见，有许多关于质数的美妙性质

唯一分解定理(也称基本算数定理)：
任意一个正整数c，将其分解为若干质数的正整数次幂的乘积，该分解方法唯一

形如：$c=p_1^{a1}*p_2^{a2}*\ldots\ldots*p_n^{an}$，$p_1\ldots p_n$均为质数

证明略

质因数分解代码：

```
t=0;
for(int i=2;i*i<=c;i++)
     if(c%i==0)
     {    p[++t]=i,a[t]=0;
          while(c%i==0)c=c/i,++a[t];
     }
if(c>1)p[++t]=c,a[t]=1;
```

唯一分解定理与LCM

将两个正整数A,B质因数分解

$A=p_1^{a1}*p_2^{a2}*\ldots\ldots*p_n^{an}$

$B=p_1^{b1}*p_2^{b2}*\ldots\ldots*p_n^{bn}$

那么：

$gcd(A,B)=p_1^{min(a1,b1)*}p_2^{min(a2,b2)*}\ldots\ldots*p_n^{min(an,bn)}$

$lcm(A,B)=p_1^{max(a1,b1)*}p_2^{max(a2,b2)*}\ldots\ldots*p_n^{max(an,bn)}$

由于$max(a,b)=a+b-min(a,b)$

所以易证明$lcm(a,b)=a*b/gcd(a,b)$

质数筛法—埃式筛法

埃式筛法

核心思想：质数的倍数一定不是质数

代码略

时间复杂度是一个调和级数

$$1+\frac{1}{2}+\cdots+\frac{1}{n}\approx \ln(n+1)+0.577\cdots$$

调和级数式子的右边一般是ln(x)+C,C是一个欧拉常数

O(nloglogn)

质数筛法—线性(欧拉)筛法

线性筛法

回顾埃氏筛法，其缺点在于一个位置可能被反复置0，浪费了时间
例如12，会被2、3置0，有没有办法让每个数只被筛去一次？
在欧拉筛法中，对于每个合数c，使得它只被作为其最小质约数的倍数筛掉

```
void Prime(int n) {
    for (int i = 2; i <= n; ++i) {
        if (!f[i]) prime[cnt++] = i; // 素数
        for (int j = 0; prime[j] <= n/i; ++j) {
            f[prime[j] * i] = true;  // 筛掉pj*i这个合数
            if (i % prime[j] == 0)  break; //核心部分
       // i%pj==0，说明pj是i的最小素因子，因此i*素数的最小素因子也是pj，在i递增的时候也会被筛

        }
    }
}
```

核心在这里

每个合数只被筛掉一次，复杂度O(n)

提高组学会线性筛基本就够了

积性函数

x,y互质，有f(x*y)=f(x)*f(y)，积性函数
任意x，y，都有f(x*y)=f(x)*f(y)，完全积性函数

欧拉函数(Euler)

就是对于一个正整数n，小于n且和n互质的正整数（包括1）的个数，记作φ(n)。

$\varphi(n)=n*(1-1/p_1)(1-1/p_2)(1-1/p_3)*(1-1/p_4)\ldots\ldots(1-1/p_n)$

其中p_1, $p_2\ldots\ldots p_n$为n的所有质因数，n是不为0的整数。φ(1)=1（唯一和1互质的数就是1本身）
欧拉函数是一个积性函数

欧拉函数的性质:

(1) p^k型欧拉函数:

若N是质数p(即N=p), $\varphi(n)=\varphi(p)=p-p^{k-1}=p-1$。

若N是质数p的k次幂(即$N=p^k$)，$\varphi(n)=p^k-p^{k-1}=(p-1)p^{k-1}$。

(2)mn型欧拉函数

设n为正整数，以φ(n)表示不超过n且与n互素的正整数的个数，称为n的欧拉函数值。若m,n互质，$\varphi(mn)=(m-1)(n-1)=\varphi(m)\varphi(n)$。

(3)特殊性质:

若n为奇数时，φ(2n)=φ(n)。

求解φ(n)

$\varphi(n)=n*(1-1/p1)(1-1/p2)(1-1/p3)*(1-1/p4)\ldots\ldots(1-1/pn)$

观察这个式子可知欧拉函数的值只与n和n的质因子种类数有关
而n是确定的，因此，只要找出N的质因子的种类数，就容易得出欧拉函数的值。

欧拉函数求解代码


```
//试除法求欧拉函数
int euler(int n){
    int res=n;
    for(int i=2;i*i<=n;i++){
        if(n%i==0){
            res=res/i*(i-1);
            while(n%i==0) n/=i;
        }
    }
    if(n>1) res-=res/n;
    return res;
}
```

求解φ(n)

推论一：如果x是质数，那么phi（x）=x-1。

证明：如果x是质数，那么除了它自身以外的所有小于x的自然数都与x互质个数为 x-1.

推论二：如果p_j是小于x的一个质因子，那么phi（x*p_j）=phi（x）*p_j 。

证明：首先我们假设 x的质因子分别是 p1、p2、....pk
那么因为 pj 是 x的一个质因子，可知pj 一定为 p1-pk 中的某一个质因子，那么把（x*pj）看作一个整体
由欧拉函数的定义得知 phi (x*pj)=(x*pj)*(1-1/p1)*(1-1/p2)*()........*(1-1/pk)(因为pj是p1到pk中的某一个质因子，所以x*pj 与x 的质因子种类是相同的)
那么进一步可得：
phi (x * pj) = pj * (x * (1-1/p1) * (1-1/p2) * ()........* (1-1 / pk)) =pj * phi (x) ;
后面这一部分恰好就是phi (x) ;

推论三：如果p_j不是小于x的一个质因子那么phi（x*p_j）=phi（x）*（p_j-1）

证明：假设一下 x的质因子分别是p1 、p2、p3....pk; 那么把（x*pj）看作一个整体他的质因子数只比原来的x多了一个就是 pj。
由欧拉函数可得 phi (x*pj) = (x*pj) * (1-1/p1) *(1-1/p2)*(1-1/p3)*......*(1-1/pk)*(1-1/pj);
整理后可得 phi (x*pj) = (1-1/p1) *(1-1/p2)*(1-1/p3)*......*(1-1/pk)) * (1-1/pj) *pj;
前面这一部分就是phi (x) ; 也就是 phi (x*pj) =phi (x) * (pj-1) ;

线性筛求欧拉1


```
int euler(int n){
        phi[1]=1;
        for(int i=2;i<=n;i++){
                if(!f[i]) primes[cnt++]=i,phi[i]=i-1;//如果i没被筛过说明是质数,根据推论一: phi[i]=i-1
                for(int j=0;primes[j]*i<=n;j++){  //筛质数顺带求一下欧拉函数值
                        f[primes[j]*i]=1;
                        if(i%primes[j]==0) //如果说pj是i 的一个质因子 利用推论二，求欧拉函数的值。
                        {
                                phi[primes[j]*i]=phi[i]*primes[j];
                                break;   //primes[j] 已经是i的最小质因子了，不再进行枚举了
                        }
                        phi[primes[j]*i]=phi[i]*(primes[j]-1);//如果pj 不是i的一个质因子，利用推论三
                }


        }
}
```

线性筛求欧拉2

根据定义简化版:

```
void euler_init(){
    euler[1]=1;
    for(int i=2;i<maxn;i++)
      euler[i]=i;                    //初始化
    for(int i=2;i<maxn;i++)
       if(euler[i]==i)               //判断是否为质数 若euler[i]!=i说明已经进行过运算了
          for(int j=i;j<maxn;j+=i)
            euler[j]=euler[j]/i*(i-1);
```

总结

数学知识在信息学考察近年来有所增强，多学点数学，对信息竞赛有好处

做题时，建议准备好草稿纸，思考问题解决的方向
数论的题目一般都要推式子，不建议一开始就去看题解

Thanks

For Your Watching

FZOI

西大附中信息学竞赛
High School Affiliated to Southwest University