数论

主讲人：数一

预备知识(假设你们已经掌握)

素数的判定(O(sqrt(n)))
素数筛
最大公约数(欧几里得算法)
分解质因数
同余运算
乘法逆元(扩展欧几里得算法)
中国剩余定理
快速幂

重点知识

费马小定理
积性函数
miller-rabin素性判断
pollard-rho分解质因数
二次剩余
原根
(扩展)离散对数
莫比乌斯反演

不准备讲但是比赛中会出现的内容

- 狄利克雷卷积
- 杜教筛
- 快速数论变换(NTT)

素数判定

- 试除法： 枚举2~sqrt(n) O(sqrt(n))
- 素数筛   O(n)+O(1)
- miller-rabin算法(今天会讲)

素数筛

- 埃拉托斯特尼筛
  可以在O(nlogn)的时间复杂度和O(n)的空间复杂度下预处理不超过n的所有素数，并且之后可以O(1)判断一个数是不是素数。

- 欧拉把上面的筛优化到了O(n)的时间复杂度，保证每一个数只被筛1次。

最大公约数

- 欧几里得算法(辗转相除法) algorithm库里面有__gcd函数，可以不用自己手写

分解质因数

- 在处理模数为**合数**时经常用的方法(先**分解质因数**->处理每一个质因子->**中国剩余定理**求解同余方程组)
  在**素数筛**时记录每一个合数第一次被筛掉时用到的素数(即**最小素因子**)
- pollard-rho算法(今天会讲)

同余运算

- 数论的基础

$$a \equiv b \pmod{n} \Leftrightarrow \text{存在一个}k\text{使得}a - kn = b$$

- 今天一切的基础

乘法逆元

同余下的"倒数"

已知a, n, 且$ax \equiv 1 \pmod{n}$，求x

$$ax \equiv 1 \pmod{n} \Leftrightarrow ax - ny = 1$$

存在条件$:\gcd(a, n) = 1$

求解算法：扩展欧几里得算法，

欧拉定理$+$快速幂(至少会一种)

扩展欧几里得算法

已知a,b,c,以及$ax-by=c$求出一组解x,y

有解条件:$\gcd(a,b)\mid c$

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

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

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

$$a\%b = a - [a/b] \cdot b$$

$$ax_1 + by_1 = bx_2 + (a - [a/b] \cdot b)y_2$$

$$ax_1 + by_1 = b(x_2 - [a/b]y_2) + ay_2$$

$$\begin{cases} x_1 = y_2 \\ y_1 = x_2 - [a/b]y_2 \end{cases}$$

中国剩余定理

$$
已知\begin{cases} x \equiv a_1 \pmod{n_1} \\ x \equiv a_2 \pmod{n_2} \\ \qquad\cdots \\ x \equiv a_k \pmod{n_k} \end{cases},求x
$$

中国剩余定理

1.计算$N = LCM\left(n_1, n_2, \ldots, n_k\right)$

2.计算$N_i = \dfrac{N}{n_i}$

3.利用$ex\gcd$计算$M_i N_i + m_i n_i = 1$

4.$x = \left(\sum_{i=1}^{k} a_i M_i N_i\right) \% N$

快速幂

在$O(\log b)$复杂度下计算$x \equiv a^b \pmod n$

考虑高精度乘法的话复杂度变为$O(\log b \log^2 n)$

利用FFT优化后是$O(\log b \log n)$

- 好了，开始上课

费马小定理

- 1640年费马提出的一个定理：
  若p是一个素数，那么对于**任意**一个整数a，
  a^p-a都是p的倍数，$a^p \equiv a \pmod p$
- 若(a,p)=1，则$a^{(p-1)} \equiv 1 \pmod p$

一个优雅的证明

- 考虑二项式系数C(p,n)，当n=0或n=p时，都是1
- 而当0<n<p时，C(p,n)=p!/(n!(p-n)!)，分子含有p，分母不含有p，因此C(p,n)%p=0
- 因此a^p=((a-1)+1)^p=sigma(i=0,p)C(p,i)(a-1)^(p-i)
  ≡(a-1)^p+1(mod p)
  ≡(a-2)^p+1+1(mod p)
  ≡(a-3)^p+1+1+1(mod p)
  ……
  ≡1+…+1+1(mod p)=a

费马小定理的推广——欧拉定理

- 模数p从素数推广到一般整数n

$$a^{\varphi(n)} \equiv 1 \pmod n$$

$\varphi(n)$为欧拉函数，

定义为不超过n的整数中与n互素的个数

$\varphi(3)=2$，$\varphi(6)=2$，$\varphi(100)=20$

欧拉定理的应用

- 求逆元

$$a^{\varphi(n)} \equiv 1 \pmod{n} \Leftrightarrow a \cdot a^{\varphi(n)-1} \equiv 1 \pmod{n}$$

$a^{\varphi(n)-1}$就是在模n下a的逆元

使用条件为$(a,n)=1$

欧拉定理的应用

- 欧拉降幂公式

$$a^b \equiv a^{b\%\varphi(n)+\varphi(n)} \pmod{n} \quad (b > \varphi(n))$$

可以把大指数转化成不超过$\varphi(n)$的指数

- 降幂，化简运算
  例题：HDU4704 (2013 多校10)

给定N，定义S_k为$x_1+x_2+\ldots+x_k=N$的正整数解的方案数，

求$\left(S_1+S_2+\ldots+S_k\right)\%1000000007$

$1<N<10^{20000}$

类似的题

- FZU1759 Super A^B mod C
  计算 A^B mod C.
  (1<=A,C<=1e9,1<=B<=1e1000000).
- BZOJ 3884上帝与集合的正确用法

$$给定p <= 10000000, 设a_0 = 1，a_n = 2^{a_{n-1}}, 求\lim_{n\to\infty} a_n \% p$$

$$即求2^{2^{2^{2^{2^{2^{2^{2^{2^{\cdots}}}}}}}}} \% p\left(无数个2\right)$$

积性函数

数论函数：定义域为正整数的函数。

积性函数：对于任意互素正整数$a,b,$都有$f(ab)=f(a)f(b)$的数论函数。

完全积性函数：对于任意正整数a,b，都有$f(ab)=f(a)f(b)$的数论函数。

例如：

常函数$f(n)=1$

单位函数$f(n)=n$

幂函数$f(n)=n^k,k$是常数

欧拉函数$\varphi(n)=\left|\{a\mid 1\le a\le n,(a,n)=1\}\right|$

因子个数$d(n)=\left|\{a\mid 1\le a\le n,(a,n)=a\}\right|$

因子k次幂和$\sigma_k(n)=\sum_{i\mid n} i^k$

莫比乌斯函数$\mu(n)=\begin{cases}1 & n=1\\ (-1)^k & n=p_1p_2\ldots p_k\\ 0 & \text{其余情况}\end{cases}$

重点

积性函数性质

$$n = p_1^{e_1} p_2^{e_2} \dots p_r^{e_r}$$

$$f(n) = f\left(p_1^{e_1}\right) f\left(p_2^{e_2}\right) \dots f\left(p_r^{e_r}\right)$$

如何求积性函数？

- 单次求积性函数 $O\left(\sqrt{n}\right)$
- 积性函数筛 $O(n)$

熟悉的复杂度...当然是熟悉的做法

单次求积性函数

- ```
  int ans=1;
  for(int p=2;p*p<=n;p++){
  	if(n%p==0){
  		//找到n的一个素因子p_i
  		int e=1;
  		while(n%p==0){
  			n/=i;
  			e++;
  		}
  		ans*=f(p,e);//计算f(p^e)
  	}
  }
  if(n>1) ans*=f(n,1);//如果有一个大于sqrt(n)的素因子
  ```
  ```

例如欧拉函数

$$n = p_1^{e_1} p_2^{e_2} \ldots p_r^{e_r}$$

$$\varphi(n) = \varphi\left(p_1^{e_1}\right)\varphi\left(p_2^{e_2}\right)\ldots\varphi\left(p_r^{e_r}\right)$$

$$\varphi\left(p^k\right) = p^k - p^{k-1} = p^k \cdot \frac{p-1}{p}$$

$$\varphi(n) = p_1^{e_1} p_2^{e_2} \ldots p_r^{e_r} \cdot \frac{p_1 - 1}{p_1} \cdot \frac{p_2 - 1}{p_2} \cdot \ldots \cdot \frac{p_r - 1}{p_r}$$

$$= n \cdot \frac{p_1 - 1}{p_1} \cdot \frac{p_2 - 1}{p_2} \cdot \ldots \cdot \frac{p_r - 1}{p_r}$$

积性函数筛

需要处理三种情况

1. f(p)，p是素数
2. f(p*i), 其中p不是i的约数，根据积性函数性质直接等于f(p)f(i)
3. f(p*i), 其中p是i的约数，需要利用函数各自的性质

积性函数筛

```
for (int i=2;i<=n;i++){
        if(!notprime[i]){
                prime[++tot]=i;
                //此处处理f(p)
        }
        for (int j=1;j<=tot && prime[j]*i<=n;j++){
                notprime[i*prime[j]]=1;
                if(i%prime[j]==0){
                         //此处处理f(i*p),p是i的约数
                        break;
                }
                else{
                        //此处处理f(i*p)=f(i)*f(p) ,p不是i的约数
                }
        }
}
```

例题

- POJ2407
  求n的欧拉函数(n<=1e9)
  POJ2478
  求n的欧拉函数前缀和(2<=n<=1e6)
  POJ1845
  求a^b的约数和，对9901取余
  (0<=a,b<=50000000)

费马小定理逆命题

- 若p是一个整数，如果对于**任意**一个与p互质的整数a，a^(p-1)%p=1，那么p是素数。
- 然而存在虽然很稀疏但是有无数个的反例(卡迈尔数)，
  最小的卡迈尔数：
  561=3*11*17
  2^560%561=1
  5^560%561=1
  7^560%561=1
  …

miller-rabin素性判断

- 1976年，Miller提出了一个基于**广义黎曼猜想**的判断一个数是素数的确定性算法
- 1980年，Rabin把上述算法改成不需要基于任何猜想的概率算法

*P.S. 2002年三位印度科学家提出了第一个确定性不基于任何猜想的通用素数判定算法AKS

miller-rabin素性判断

二次探测定理：

若n是一个素数，且$0 < x < n$，则方程$x^2 \equiv 1 \pmod n$的解为$x = 1$或$x = n-1$

引理：

设一个奇素数$n = 2^s \cdot d + 1$，其中d为奇数，s为正整数，对于任意a要么满足$a^d \equiv 1 \pmod n$

要么满足$a^{2^r \cdot d} \equiv -1 \pmod n$对于$0 \le r \le s-1$都成立。

二次探测定理引理的逆否命题：

设大于2的一个奇数$n = 2^s \cdot d + 1$，其中d为奇数，

s为正整数，若我们能找到一个a满足$a^d \neq 1 \pmod{n}$

且$a^{2^r \cdot d} \neq -1 \pmod{n}$对于$0 \le r \le s-1$都成立，

那么n不是素数。

如何选择底数？

- 随机？
  概率论分析可以得到每次随机检测的错误率为25%
- 执行k次测试后，检测错误的概率为

$$\frac{1}{4^k}$$

如何选择底数？

- 对于64位以内的奇数 $n < 18{,}446{,}744{,}073{,}709{,}551{,}616 = 2^{64}$,只需要选取 a = 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 以及37(40以内的素数)即可100%确定素数。

复杂度？

设测试次数为k，要测试的数为n，

那么考虑朴素高精度乘法后的复杂度为$O\left(k\log^3 n\right)$,

使用FFT来实现高精度乘法后的复杂度为$O\left(k\log^2 n\right)$

Pollard-rho算法

- 1975年由John Pollard提出的一个用于求给定合数的最小质因子的算法。

Pollard-rho 算法思路

若存在两个整数x_1, x_2，且n不是$x_1 - x_2$的约数，

则$x_1 - x_2$的其中一个因子$p = \gcd\left(x_1 - x_2, n\right)$

也是n的一个因子。

怎么找到这样的x1,x2?

1.随机选取一个x_1

2.利用一个设计好的函数$f(x)$计算$x_2 = f(x_1)$,

通常使用$f(x) = x^2 + c$(c是一个常数)

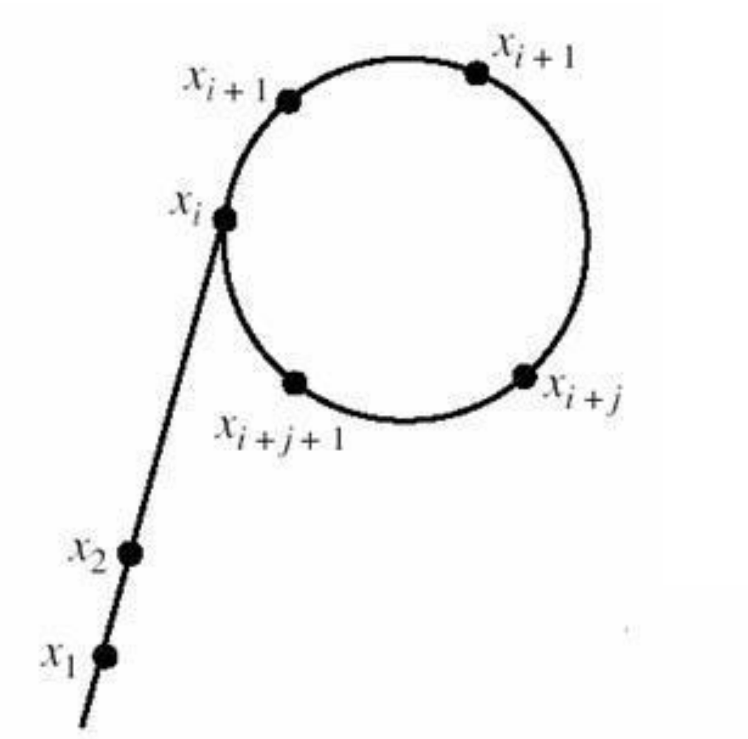
3.计算$d = \gcd(x_1 - x_2, n)$,

若d不是1，则d是n的一个因数

否则，令$x_1 = x_2$，返回步骤1

为啥是rho？

- 我们输出每次迭代的x[i]，可以发现从某一个i开始，会陷入一个循环



rho:

ρ

找到一个因子p后

- 对p和n/p递归使用Pollard-rho算法，直到都是素数。

POJ1811

- 给出一个N(2<=N<2^54) 如果是素数，输出Prime，否则输出N的最小素因子。

二次剩余

- 什么是二次剩余？

已知$x^2 \equiv n \pmod{p}$,

则称x为n在模p意义下的二次剩余

(可以看成是模p意义下的$\sqrt{n}$)

二次剩余存在的条件

- 欧拉判定准则(1748年)

p是一个奇素数，a与p互质，

那么$x^2 \equiv a \pmod{p}$有解当且仅当$a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$

这样的a称为模p意义下的完全平方数

如何求二次剩余

- Cipolla算法

1.找到一个$0 \le a < p$，使得$a^2 - n$不是一个模p意义下的完全平方数(用欧拉准则判断)

2.则$x = \left(a + \sqrt{a^2 - n}\right)^{\frac{p+1}{2}}$为其中一个二次剩余

怎么计算这个？

另一个？

小试牛刀

- URAL1132
- 给定a,n，求x*x%n=a的解，若无解，输出“No root”,否则把根从小到大输出。

离散对数

- 什么是对数？

已知$a^x = b$的a,b，那么$x = \log_a b$，称为以a为底b的对数

- 什么是离散对数？
  在模n下的对数运算

已知$a^x \equiv b(\text{mod} n)$的$a,b,n$,那么$x \equiv \log_a b(\text{mod}\, n)$,

称为在模n下以a为底b的对数

怎么求离散对数？

- 这是一个很经典的工业问题，因为没有O(log c)复杂度的解法，所以可以作为公钥加密算法——DH密钥交换算法
- 竞赛中只需要掌握O(sqrt(c))的算法即可
- 求解算法
- Baby Step Giant Step(竞赛中常用)
- Pollard-rho (对，又是这个人)

Baby Step Giant Step

- 一个用空间换时间的算法(hash表判重)
- 一个中间相遇法的算法(节省一半搜索状态)

算法思想

假设我们解的方程$a^x \equiv b \pmod{n}$的结果是x

把x写成$x = im + j \; (0 \le i, j < m)$，并且取$m = \left\lceil \sqrt{n} \right\rceil$

$$(a^m)^i a^j \equiv b \pmod{n} \Leftrightarrow a^j \equiv b\left(a^{-m}\right)^i \pmod{n}$$

baby step

hash

giant step

复杂度

- 因为i,j都是不超过m=sqrt(n)的数，因此整体复杂度是O(sqrt(n))的

应用条件？

- n必须是素数。
- 如果n不是素数会出现什么问题？

ex-baby step giant step

- 分析
- n不是素数之所以不成立是因为(a,n)!=1导致逆元不存在。

设$g=\gcd(a,n)$

$a^x-kn=b$

当g不能整除b时，必然无解

两边同时除以g

$$\frac{a^x}{g}-\frac{kn}{g}=\frac{b}{g}\Leftrightarrow\frac{a}{g}a^{x-1}\equiv\frac{b}{g}\left(\mathrm{mod}\,\frac{n}{g}\right)\Leftrightarrow a^{x-1}\equiv\frac{b}{g}\cdot\left(\frac{a}{g}\right)^{-1}\left(\mathrm{mod}\,\frac{n}{g}\right)$$

手算一下？

$$6^{x} \equiv 8 \pmod{16}$$

$$6^x \equiv 8 \pmod{16}$$

$$\gcd(6,16) = 2$$

$$\frac{6}{2} \cdot 6^{x-1} \equiv \frac{8}{2} \left(\bmod \frac{16}{2} \right)$$

$$6^{x-1} \equiv 4 \cdot 3^{-1} \equiv 4 \pmod{8}$$

$$6^{x-2} \equiv 2 \cdot 3^{-1} \equiv 2 \pmod{4}$$

$$6^{x-3} \equiv 1 \cdot 3^{-1} \equiv 1 \pmod{2}$$

$$x - 3 = 0$$

$$x = 3$$

题目

- POJ2417
  找到最小的L，满足$B^L == N \pmod P$，其中P是素数。

题目

- HDU2815 MOD Tree
  求最小的D，满足K^D=N(mod P)，不存在输出” Orz,I can’t find D!”
  P.S. 注意这里的can右上角的那一撇是全角的引号……为了减少不必要的WA，请直接复制题面…

原根(Primitive Root)

给定一个数n，若存在一个与n互素的a，使得$a^i\left(i=0,1,\ldots,\varphi\left(n\right)\right)$在模$n$下两两不同，那么称$a$是$n$的一个原根

举个栗子

- 在模7的情况下，3是一个原根，因为

$$3^0 \equiv 1$$

$$3^1 \equiv 3$$

$$3^2 \equiv 2$$

$$3^3 \equiv 6$$

$$3^4 \equiv 4$$

$$3^5 \equiv 5$$

$$3^6 \equiv 1$$

合数也有原根

$\varphi(14) = 6$

x	x^2	x^3	x^4	x^5	x^6
1	1	1	1	1	1
3	9	13	11	5	1
5	11	13	9	3	1
9	11	1	9	11	1
11	9	1	11	9	1
13	1	13	1	13	1

3和5是14的原根

哪些模数有原根？

- $n=1,2,4,2p,p^r$
  其中p是奇素数

如何求原根？

1. 枚举….至少是O(n)
2. 利用原根的性质

1.计算欧拉函数$\varphi(n)$

2.对$\varphi(n)$分解质因数$\varphi(n)=p_1^{e_1}\cdot p_2^{e_2}\cdot\ldots\cdot p_r^{e_r}$

3.枚举$a=2\sim n-1$，用下面的方法检验是否为原根

4.若$a^{\frac{\varphi(n)}{p_1}}$，$a^{\frac{\varphi(n)}{p_2}}$，…，$a^{\frac{\varphi(n)}{p_r}}$中至少出现一个1，

则a不是n的原根，若都不是1，则a是n的原根。

计算欧拉函数复杂度为$O(\sqrt{n})$

单次检验的复杂度为$O(\log^2 n)$

一般原根都很小

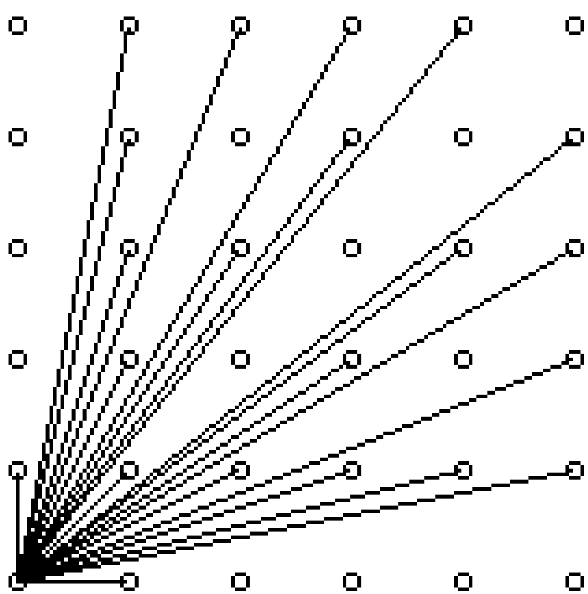
原根的一些性质

- 一个数n如果有原根，那么有phi(phi(n))个
- 高斯证明了：
- 一个数n的全体原根乘积模n余1
- 一个数n的全体原根总和模n余μ(n-1)(莫比乌斯函数)

- POJ1284
- 给出一个p(3<=p<65536)，求这个数有多少个原根
- HDU4992
- 给出一个n(2<=n<1000000)，求所有原根。

莫比乌斯反演

- 先看这道题：
  POJ3090
  有一个n*n的二维格点，问在原点(0,0)处能看到多少个格点？(n<=1000,1000组数据)



n=5的情况，答案是21，
如左图的21根线

$$\left|\left\{(x,y) \mid 1 \le x, y \le n, \gcd(x,y)=1\right\}\right|$$

$$3+2\sum_{i=1}^{n}\varphi(i)$$

三维怎么办？

- SPOJ – VLATTICE
  有一个n*n*n的三维格点，问在原点(0,0,0)处能看到多少个格点？(n<=1000000,50组数据)

$$\left|\left\{(x,y,z) \mid 1 \le x,y,z \le n, \gcd(x,y,z)=1\right\}\right|$$

莫比乌斯反演！

什么是莫比乌斯反演？

我们如果需要求一个函数$g(n)$，但是这个函数不好求。

我们又发现$f(n)=\sum_{d|n} g(d)$很好求。

那么我们就可以通过求f，来间接求g：$g(n)=\sum_{d|n}\mu(d)f\left(\frac{n}{d}\right)$

$$\text{莫比乌斯函数}\mu(n)=\begin{cases}1 & n=1\\(-1)^k & n=p_1p_2\dots p_k\\0 & \text{其余情况}\end{cases}$$

一些有趣的性质

$$\sum_{d|n}\mu(d)=\begin{cases}1, n=1\\0, n\neq 1\end{cases}$$

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

两种形式

- 本质：在自然数中的容斥

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

$$\Downarrow$$

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

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

$$\Downarrow$$

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

证明就一行

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

令 $\frac{n}{d}=k$

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

函数定义

交换求和顺序

莫比乌斯性质

另一种形式(这种用得比较多)

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

$$\sum_{n|d}\mu\left(\frac{d}{n}\right)f(d),\text{令}\frac{d}{n}=k,\text{则}$$

$$\sum_{k=1}^{\infty}\mu(k)f(nk)=\sum_{k=1}^{\infty}\left(\mu(k)\sum_{nk|t}g(t)\right)=\sum_{n|t}\left(g(t)\sum_{k|\frac{t}{n}}\mu(k)\right)=g(n)$$

函数定义

交换求和顺序

莫比乌斯性质

设$g(m)=\left|\{(x,y,z)\mid 1\le x,y,z\le n,\gcd(x,y,z)=m\}\right|$

我们要求的答案就是$g(1)+6$。(3个轴,3个面平分线)

$$f(m)=\sum_{m|d}g(d)=\left|\{(x,y,z)\mid 1\le x,y,z\le n,m\mid \gcd(x,y,z)\}\right|$$

$f(m)$的意义为在点阵中,$\gcd(x,y,z)$是m的倍数的点数。

$$f(m)=\left\lfloor\frac{n}{m}\right\rfloor\left\lfloor\frac{n}{m}\right\rfloor\left\lfloor\frac{n}{m}\right\rfloor$$

于是进行反演:

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

应用

- 于是可以解决大量”有多少组(x,y),gcd=k”的统计问题
  HDU1695
  给定a,b,c,d,k,求有多少组无序整数对(x,y)，满足a<=x<=b,c<=y<=d,gcd(x,y)=k
  a=1,c=1
  1<=b,d<=100000
  0<=k<=100000

- HDU5212

给定n个数$a_1, a_2, \ldots a_n$,

计算$\sum_{i=1}^{n}\sum_{j=1}^{n}\gcd(a_i, a_j)\cdot(\gcd(a_i, a_j)\text{-}1)$

- https://vjudge.net/contest/176263
- 或者搜索 “2017 Summer Training（精英班） Day4”
- 请各位同学用手机或者电脑先进去contest，登录好自己的账号，然后停留在输入密码页面。
- 前三个找到密码并且进去提交代码的(以status的runid为准)可获得公仔一个，不需要AC

放松一下

1.最小的素数是多少？

2.设上题答案为a，请问$ax+4y=2017$有多少个整数解$\left(x,y\right)$?

3.设上题答案为b，求最小的x，满足$3^x\equiv b\left(\mathrm{mod}129140163\right)$.

4.设上题答案为c，求$\left(b+ac\right)\%26$。

5.设上题答案为d，求d在模$\left(a^5-1\right)$下的最小正逆元。

6.设上题答案为e，把$abcde$写在一起得到一串数字

就是今天题目的密码。