7．数论基础

张桄玮(gwzhang@cug.edu.cn)

郑州一中(Legacy)

2024-08-10

整除关系

基本定义与性质

- 整除记号: 把因子抓到前面(Def1.1)
- 最大公约数(Def1.2)
  - 可以用作化简
  - 特殊情况: 注意 0 的情形
- 最小公倍数(Def1.3)

最小公倍数和最大公约数有什么联系?

- 常见的证明方法: 表示为$n = mk$.

求最大公约数(Euclid 算法)

- 大家都知道$\gcd(a,b) = \gcd(b, a \bmod b)$

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

证明这件事就要请出扩展 Euclid 算法.

- 解答的同时还可以帮助得到一个不定方程的一组解!

除法表达式(div-prob)

问题：给出一个这样的除法表达式:
$X_1/X_2/X_3/X_4/X_5/X_6..../X_k$,

正常的除法表达式是从左往右计算的, 但是我们可以向其中添加括号, 以改变它的运算顺序, 如 1/2/1/2=1/4, (1/2)/(1/2)=1 ;

那么， 给定一个除法表达式， 能否通过添加括号使它的值为正整数？

除法表达式(div-prob)

考虑整除的性质, 最后必须为$X_1 X_2 \ldots \frac{X_k}{X_2}$的形式 – 即

- 一堆放在分子, 一个放在分母.
- 如果分母放得多的话反而会变得更 "不整".

约分: 每次约分掉X_i和X_2的$\gcd$.

- 整数 $\Longleftrightarrow X_2 = 1$.

扩展 Euclid 算法

问题: 求出满足$ax+by=\gcd(a,b)$的(x,y)的解答.

求一组特解:

```
int exgcd(int a, int b, int &x, int &y){
    if(b==0) {x=1,y=0;return a;}
    int r = exgcd(b,a%b,x,y);
    tie(x,y) = make_tuple(y,x-(a/b)*y);
    return r;
}
```

证明方法: 数学归纳法;

假设有了一个解(x_1,y_1). 如何得到其他的解?

- 取另一组解(x_2,y_2), you $ax_1+by_1=ax_2+by_2=\gcd(a,b)$

扩展 Euclid 算法

- 以a, b提取系数: $a(x_1 - x_2) = b(y_2 - y_1)$, 两边同时除以$g := \gcd(a, b)$
- 得到$a'(x_1 - x_2) = b'(y_2 - y_1)$. ($a' = \frac{a}{g}, b' = \frac{b}{g}$).

> 性质 *0.1.*: 设$a, b, c \in \mathbb{Z}$, 一旦找到一组$ax + by = c$的整数解x_0, y_0, 它的其他解可以写作$(x_0 + kb', y_0 - ka')$.

由$ax + by = \gcd(x, y)$得到$ax + by = c$:

- 例如$6 \times (-2) + 15 \times 1 = 3$, 同时乘 3 即可得到$6 \times (-6) + 15 \times 3 = 9$.

P1516 青蛙的约会

问题：我们把这两只青蛙分别叫做青蛙 A 和青蛙 B，并且规定纬度线上东经 0 度处为原点，由东往西为正方向,单位长度 1 米，这样我们就得到了一条首尾相接的数轴。设青蛙 A 的出发点坐标是 x，青蛙 B 的出发点坐标是 y。青蛙 A 一次能跳 m 米，青蛙 B 一次能跳 n 米，两只青蛙跳一次所花费的时间相同。纬度线总长 L 米。现在要你求出它们跳了几次以后才会碰面。

P1516 青蛙的约会

要求:

$$
\begin{aligned}
(x+km) \bmod l &= (y+kn) \bmod l \\
\Rightarrow (x+km)-(y+kn) &= lz, z \in \mathbb{Z} \\
x-y+k(m-n)-lz &= 0
\end{aligned}
$$

谁是常数, 是是变量?

$$
\begin{aligned}
k(m-n)-lz &= -(x-y) \\
-k(n-m)-lz &= -(x-y) \\
k(n-m)+lz &= x-y
\end{aligned}
$$

要解的是(k,l). 使用 `exgcd` 即可.

类似练习: P1082; P3811

整除与求和作用的结果

- 因数的共轭性质
- 求和记号的交换

质数与合数

基本定义

- 质数
- 合数

唯一分解定理与每个数的坐标

- 唯一分解定义(证明)
- 整数坐标(无穷维空间);

UVA10375 选择与除法

考虑每个素数的指数位置, 最后求解.

CF582A GCD Table

重要观察: 两个数的gcd一定不大于两个数.

- $\gcd(a, b) \leq \min(a, b)$

找到第一大和第二大的数

- 将原来的$n \times n$从大到小排序.
- 最大的两个数是a_1, a_2.

第三大的数呢?

- 不一定在原数列里面, 可能是最大的那两个的gcd.
- 把$\gcd(a_1, a_2)$去掉, 也把$\gcd(a_2, a_1)$去掉.
- 威胁解除!!

每次把数表中的最大值取出来，它是原数列中的数。然后，把它与之前被取出来的 gcd都从数表中剔除。如此循环往复, 直到取满n个数或数表被取空为止。

求和式的交换

求和式的一种做法: 枚举答案

a) Iverson 括号与求和记号

- 基本定义
- 两个 Iverson 括号的合并与拆分

b) 枚举答案: 下取整为例

- 考虑答案的值
- 下取整的性质

CQOI2007 余数求和

$$\sum_{i=1}^{n} k \bmod i = \sum_{i=1}^{n} k - i\left\lfloor \frac{k}{i} \right\rfloor$$
$$= nk - \sum_{i=1}^{n} i\left\lfloor \frac{k}{i} \right\rfloor$$

- 然后枚举$\left\lfloor \frac{k}{i} \right\rfloor$的答案来做(假定$k = 5$)

```
   i         1  2  3  4  5  6  7  8  9  10
ceil(5/i)    5  2  1  1  1  0  0  0  0  0
```

枚举左端点算右端点:

$$t \le \frac{k}{i} < t + 1$$
$$\left\lceil \frac{k}{t+1} \right\rceil < i \le \left\lfloor \frac{k}{t} \right\rfloor$$

UVA12716 GCD=XOR

回顾 xor 的性质: 不进/退位的二进制加/减法

- $a - b \le a \oplus b \le a + b$;
- $a \oplus b = c \Longrightarrow a \oplus c = b$;

要求

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

可以枚举答案, 考察gcd的值记为d:

$$\sum_{d=1}^{n}\sum_{d\backslash i}[\gcd(i, i \oplus d) = d].$$

但是还是会 TLE, 考虑gcd的大小关系.

质数筛法

- Eratosthenes 筛法

```
void Prime(){
  for(int i=2;i<=N;i++){
    if(notp[i]==0){
      prime[++cnt] = i;
      for(int j=2*i; j<=N;j+=i){
        notp[j] = 1;

      }
    }
  }
}
```

- Euler 筛
  - 每个数只会被其最小的质因数筛掉
  - 和唯一分解定理打很好的配合

质数筛法


```
void Prime(){
  for(int i=2;i<=N;i++){
    if(notp[i]==0){
      prime[++cnt] = i;
      for(int j=2*i; j<=N;j+=i){
        notp[j] = 1;

      }
    }
  }
}
```

获得一个区间里面的质数.

有理数是可数的吗?

是的. 可以使用 Stern-Brocot 树说明.

模运算

基本运算规则

- 定义与记号
- 加, 减(注意先加模数), 乘法
  - 防止溢出: 增加`-sanitizer=undefined`
- 快速幂

P1226 快速幂取余

问题：计算$ab \bmod k$.

除法与逆元

被约去的数和当前的模数互质的情况下, 才可以约去.

UVA11582

问题：输入两个非负整数 a, b 和正整数 $n(0 \leq a, b < 2^{64}, 1 \leq n \leq 10^3)$, 求 $f\left(a^b\right) \bmod n$ 的值, 其中 $f_0 = 0, f_1 = 1, f_{i+2} = f_i + f_{i+1}(i \geq 0)$ 。

UVA11582

- 有取模: 考虑是不是有循环节? (Yes!)

深入理论:

- 一个数 $\bmod n$, 有 n 种可能 $(\{m : 0 \leq m < n\})$. 一个二元组 $(f_i \bmod n, f_{i-1} \bmod n), i >$ 有 $n \times n$ 种.
- 若一个二元组 $\left(f_q, f_{q-1}\right) = \left(f_p, f_p - 1\right) 0 < q, p$, 那它们之后的所有数也会出现循环.

P2613 有理数取余

- 类似于求乘法

经常出现在为了避免小数点后的精度的问题.

解线性同余方程组

(ex)crt: 将多个式子两两合并, 化为一个.

数论函数

简介

- 定义域和值域通常在整数上面

Euler 函数

- 定义
- 求法 1: 考虑归纳的方法

```
int prime[MAXN], phi[MAXN], cnt = 0; bool notp[MAXN];
int main(){
  for(int i =2;i<=N;i++){
    if(notp[i]==0){
      prime[++cnt] = i; phi[i] = i-1; // 是质数
    }
    for(int j=1;i*prime[j]<=N&&j<=cnt;j++){
      notp[i*prime[j]] =1;
      if(i % prime[j] == 0) {
        phi[i*prime[j]] = phi[i] * phi[prime[j]];
        break;
      }else{phi[i*prime[j]] = (j-1)*phi[i];}
    }
  }
}
```

积性函数

- 定义

Mobius 函数

- 发明的动机

```
for (int i = 2; i <= N; i++) {
    if (!notp[i]) {  // i is a prime number
        prime[++cnt] = i;
        mu[i] = -1;
    }
    for (int j = 1; j <= cnt && i * prime[j] <= N; j++) {
        notp[i * prime[j]] = true;

        if (i % prime[j] == 0) {
            mu[i * prime[j]] = 0;
            break;
        } else {
            mu[i * prime[j]] = mu[i] * mu[prime[j]];
        }
    }
}
```

P2158 仪仗队

- 左下角当做坐标轴的原点(0,0)
- 被看到的人满足$\gcd(x, y) = 1$成立.

$$
\begin{aligned}
&\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}[\gcd(i, j) = 1] \\
&= 2\sum_{i=1}^{n-1}\sum_{j=1}^{i-1}[\gcd(i, j) = 1] + 3 \\
&= 2\left(\sum_{i=1}^{n-1}\varphi(i)\right) + 3 \\
&= 2\sum_{i=1}^{n-1}\varphi(i) + 3.
\end{aligned}
$$