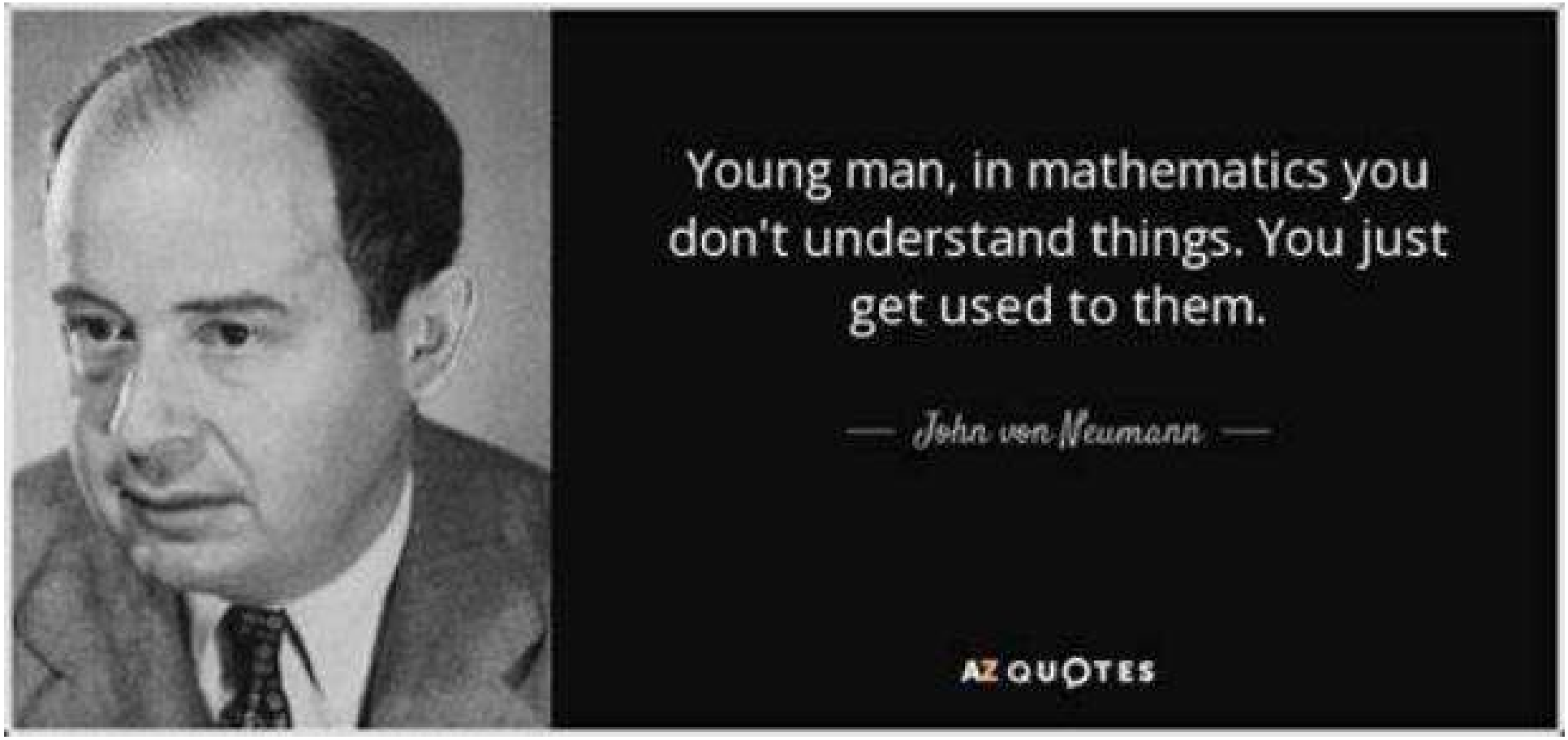
数学知识

肖然
北京大学附属中学 信息与通用技术中心
2017.09
xiaoran@pkuschool.edu.cn


Young man, in mathematics you don't understand things. You just get used to them.
— John von Neumann —
AZ QUOTES

目录

- 数论
  - 除法定理
  - 质数筛法
  - exGCD，欧拉函数，欧拉定理/费马小定理，威尔逊定理
  - 唯一分解
  - 除法分块
  - 同余最短路
  - 中国剩余定理
- 组合数学
  - 杨辉三角/二项式定理
  - 多重集排列/多重集组合
  - 错排列/圆排列
  - 鸽巢原理
  - 容斥原理
  - Catalan数
  - 组合恒等式

除法定理

数论

- 研究【整数】的性质。
- 计算机中只能使用二进制组成整数，实数运算在底层实际都是整数的运算。

- 整数集 $Z = \{..., -2, -1, 0, 1, 2, ...\}$
- 自然数集 $N = \{0, 1, 2, ...\}$

整除

- 存在整数k，使得 $a = kd$，则有 $d|a$（d整除a）
- 任何整数都整除0

- 如果 $d|a$并且$d > 0$，则称d是a的**约数**，a是d的**倍数**。
  一个整数a的约数最小为1，最大为a。

除法定理

- 对任意整数a和任意正整数n，存在唯一整数 q 和 r：
  - $a = qn + r, 0 \leq r < n$
  - $q = a/n$ (向下取整)称为除法的商
  - $r = a\%n$ 称为除法的余数。

- $n|a$当且仅当 $a\%n = 0$
- $n|a$读作“n整除a”“a被n整除”

整除的一些性质

- $d|x, x|y \rightarrow d|y$(传递性)
  - $d|x, d|y \rightarrow d|(x \pm y)$
    - $d|x \rightarrow d|kx$

公约数

- 若 $d|x$ 并且 $d|y$, 则d是x, y的公约数.
- 公约数整除线性组合:
  - $d|x, d|y \rightarrow d|(ax+by)$, 其中 $a, b \in Z$, $(ax+by)$称为x, y的线性组合

- x, y的公约数中最大的那个称为最大公约数$gcd(x, y)$

gcd的性质

- 裴蜀定理：
  - gcd(x, y)是x, y线性组合集合中的最小正元素.
  - 怎么证明？
    - 设最小线性组合 s = ax + by
    - s ≥ gcd(x, y): gcd整除线性组合
    - s ≤ gcd(x, y): s是x, y公约数，即 s|x 且 s|y

- d|x, d|y → d|gcd(x, y) [公约数整除gcd]
- gcd(xn, yn) = n·gcd(x, y)
- 如果 n|xy 并且 gcd(n, x) = 1，则 n|y

除法定理-总结

- 除法定理：
  - 对任意整数a和任意正整数n，存在唯一整数 q 和 r：

$$a = qn + r, 0 \leq r < n$$

- 裴蜀定理：
  - gcd(x, y)是x, y线性组合集合中的最小正元素

P8255 [NOI Online 2022 入门组] 数学游戏

- 给x, z，求最小的整数y满足 $x \times y \times \gcd(x, y) = z$

若 $x \nmid z$，则一定无解。

令 $x = d \times a$，$y = d \times b$，其中 $\gcd(a, b) = 1$。那么有 $\gcd(x, y) = d$。则 $z = x \times y \times \gcd(x, y) = d^3 \times a \times b$。

我们知道 x 与 z，那么可以求出 $\frac{z}{x} = d^2 \times b$。由于 $\gcd(a, b) = 1$，所以 $\gcd(\frac{z}{x}, x^2) = \gcd(d^2 \times b, d^2 \times a^2) = d^2$。若 $\gcd(\frac{z}{x}, x^2)$ 不是完全平方数，则无解。

将 $\gcd(\frac{z}{x}, x^2)$ 开方即可得到 d。那么：

P6476 [NOI Online #2 提高组] 涂色游戏

你有 10^{20} 个格子，它们从 0 开始编号，初始时所有格子都还未染色，现在你按如下规则对它们染色：

1. 编号是 p_1 倍数的格子（包括 0 号格子，下同）染成红色。
2. 编号是 p_2 倍数的格子染成蓝色。
3. 编号既是 p_1 倍数又是 p_2 倍数的格子，你可以选择染成红色或者蓝色。

其中 p_1 和 p_2 是给定的整数，若格子编号是 p_1 或 p_2 的倍数则它必须要被染色。在忽略掉所有未染色格子后，你不希望存在 k 个连续的格子颜色相同，因为你认为这种染色方案是无聊的。现在给定 p_1, p_2, k，你想知道是否有一种染色方案不是无聊的。

质数筛法

质数

- 只能被1和自己整除的数，称为质数
- 质数有无穷多个
  - 素数定理：当N充分大时，1~N范围内的质数个数~N/lgN；
  - 可以这样理解，在[1,N]范围内随机抽取一个数，抽到质数的概率为1/lgN
  - 质数分布：在某个正整数N附近能够在log(N)期望时间内发现一个质数

- 如何判断一个整数n是否为质数：2到√n试除，发现约数则不是质数
- 复杂度O(√n)
- 约数成对出现（完全平方数除外）

2到√n试除

```
bool isprime(int x){
    if(x==1) return 0;//特判 x = 1
    for(int i=2;i*i<=x;++i){//不要写成 i<=sqrt(x)
        if(x%i==0) return 0;
    }
    return 1;
}
```

复杂度$O(\sqrt{n})$，1s内能够判断 1e16 范围内的N是否为质数

生成一个区间内的素数 – 筛法

- 从小到大，依次将每个质数的倍数删去
- 若遍历到一个数时其没有被删去，则可知这个数是质数
- 最终留下的全是质数。

- 复杂度O(NloglogN)，即质数倒数和，近似线性
- 更精细的做法：欧拉筛法O(N)

Eratothenes筛法

```
bool isComp[MAXN];//记录是否为合数
isComp[1] = 1; //特判1为合数
for(ll i=2;i*i<=N;i++) {
    if(isComp[i]==0)
        for(ll j=i*i;j<=N;j+=i)
            isComp[j]=1;
}
//注意需要开ll
//近似线性，非常快
```

P7960 [NOIP2021] 报数

- 设 p(x) 表示 x 的十进制表示中是否含有数字 7，若含有则 p(x) = 1。则一个正整数 x 不能被报出，当且仅当：
  - 存在正整数 y 和 z 使得 x = yz 且 p(y) = 1
- 已知上一个数报出了 x，快速算出他下一个数要报多少，多测
- T = 2e5, x = 1e7

- 从小到大，依次将满足p(y)=1的y的倍数删去（若遍历到一个数时，其没有被删去，则拆位计算p(y)）
- 需要筛的y有1e6个，它们的倒数和为1.6左右
- 筛去这些y的倍数，然后预处理答案即可
- 注意特判 ans[n=1e7]

```
int n = 1e7;
int tot = 0, sum = 0;
for(int i=1;i<=n;i++){
    if(vis[i]) continue;
    if(chk(i)){//拆位
        ++tot;
        for(int j=i;j<=n;j+=i){
            vis[j] = 1;
            sum++;
        }
    }
}
//tot = 923655
//sum = 16750304
```

CF937B Vile Grasshoppers

- 给你两个数 p, y
- 求去掉2至y中所有2至p的数的倍数后剩下的最大值
- 没有则输出 -1

- $p \leq y \leq 1e9$

欧拉筛法O(N)

- Eratothenes筛法 – 复杂度O(NloglogN)
  - 有些数会被重复筛，例如6会被2、3都筛一遍
  - 如果能严格保证每个数只被筛一次，复杂度为O(N)

- 原理：对每个合数n，用n的最小质因子p_j将其筛去
  - 考虑合数$n = p_j \times i$，如何保证 n 仅被 p_j 筛去呢？
  - 2筛去4、6、8、10...
  - 3筛去9、15、21、27...
  - 5筛去25、35、55、65...
  - 7筛去49、77、91、119

i \ p_j	2	3	5	7
2	4			
3	6	9		
4	8			
5	10	15	25	
6	12			
7	14	21	35	49
8	16			
9	18	27		
10	20			
11	22	33	55	77

1. 第i行时，筛去当前已经发现的质数p_j与 i 的乘积，并保证**质数p_j是 i × p_j的最小质因子**
2. 当发现p_j|i时，则不应该再用 $> p_j$ 的质数 p_k 来筛 i × p_k，因为此时i × p_k 的最小质因子已经不再是p_k（而是p_j）

```cpp
bool vis[MAXN];//合数标记
int prime[MAXN], cnt = 0;//质数表
void euler(){
    vis[1] = 1;
    n = 1e7;
    for(int i=2;i<=n;i++){
        if(!vis[i]) prime[++cnt] = i;//发现新质数
        for(int j=1;j<=cnt && i*prime[j]<=n;j++){
            vis[i*prime[j]] = 1;//prime[j]是i*prime[j]的最小质因子
            if(i%prime[j]==0) break;
        }
    }
}
```

欧拉函数

- 对于正整数n，欧拉函数$\varphi(n)$的值是“小于等于n的数中，与n互质的数的个数”
- 容斥：
  - $\varphi(n) = n \times (1 - 1/p_1) \times (1 - 1/p_2) \ldots \times (1 - 1/p_t)$
  - 其中$p1, p2 \ldots p_t$为n的所有质因数

- 将n质因数分解后求解$\varphi(n)$，复杂度$O(\sqrt{n})$
- 使用跳跳法（类似筛法思路），在$O(nloglogn)$时间内求出1~n范围内的所有φ值
- 欧拉筛，复杂度$O(n)$

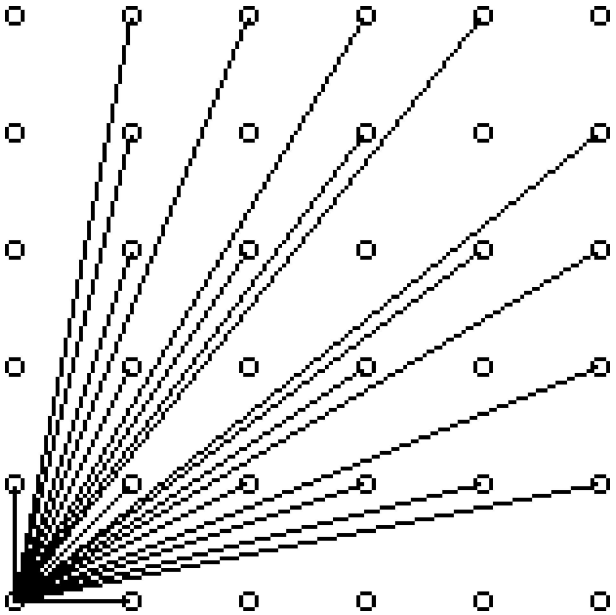
欧拉函数$\varphi(x)$

- $\varphi(x)$函数满足以下性质：
  - $\boldsymbol{\varphi(n) = n\prod_{p|n}(1-\frac{1}{p})}$，$\boldsymbol{p}$为$\boldsymbol{n}$的所有质因子
  - $\boldsymbol{\varphi(p) = p-1}$，$\boldsymbol{p}$为质数
  - $\boldsymbol{\varphi(p^k) = p^k - p^{k-1}}$
  - $\boldsymbol{\varphi(mn) = \varphi(m)\varphi(n)\frac{d}{\varphi(d)}}$，$\boldsymbol{d}$为$\gcd(\mathrm{m},\mathrm{n})$
  - 若$\gcd(\mathrm{m},\mathrm{n}) = 1$: $\boldsymbol{\varphi(mn) = \varphi(m)\varphi(n)}$

- 考虑 $\boldsymbol{n}$ 的最小质因子 $\boldsymbol{p_j}$ 满足 $\boldsymbol{n = p_j \times i}$:
  - 若$\boldsymbol{p_j|i}$: $\varphi(n) = p_j \times \varphi(i)$
  - 否则：$\varphi(n) = (p_j - 1) \times \varphi(i)$

P2158 [SDOI2008] 仪仗队

- 仪仗队是由学生组成的 N × N 的方阵，C 君希望你告诉他队伍整齐时能看到的学生人数



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

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

$$=2\sum_{i=1}^{n}\sum_{j=1}^{i}[gcd(i,j)=1]-1$$

$$=2\sum_{i=1}^{n}\phi(i)-1$$

预处理 $\phi(d)$ 前缀和，$O(1)$ 解决。

质数筛法 – 总结

- 质数定理
- 埃氏筛法/跳跳法
- 欧拉筛法
- 欧拉函数

唯一分解

唯一分解定理（算术基本定理）

- 任何大于1的正整数n仅能以**唯一方式**分解为有限质数的乘积：
  - $n = {p_1}^{e1} \cdot {p_2}^{e2} \cdots {p_t}^{et}$
  - 其中p_i为质数，$p_1 < p_2 < \cdots < p_t$，e_i为正整数

- 算术基本定理是一条非常基本和重要的定理，它把对自然数的研究转化为对其最基本的元素——素数的研究。
- t, $\sum e_i$都是 logn 量级的

求n的质因数分解（唯一分解）

- 2到√n试除，如果遇到n的约数就循环除，记下约数和除的次数
- √n之内若未将n除尽，则剩余的为n的最大质因数（至多有一个大于√n的质因数）

- 复杂度O(√n)

质因数分解

```
for(int i=2;i*i<=n;i++){// 2到√n试除
    int ei = 0;
    while(n%i==0) { // 遇到约数就除尽
        n /= i;
        ei++;
    }
    if(ei) cout<<i<<" "<<ei<<'\n';
}
// 若未将n除尽，则剩余的为n的最大质因数(大于sqrt(n))
if(n!=1) cout<<n<<" "<<1<<'\n';
```

复杂度O(√n)
i++执行O(√n)次
ei++执行O(logn)次

约数的唯一分解

- 若n的唯一分解为：
  - $n = {p_1}^{e1} \cdot {p_2}^{e2} \cdots {p_t}^{et}$, 其中$e_i > 0$
- 则n的约数d可表示为：
  - $d = {p_1}^{v1} \cdot {p_2}^{v2} \cdots {p_t}^{vt}$, 其中$e_i >= vi >= 0$

- n的约数个数：$\prod(e_i + 1)$
- n的约数和：$\prod(1 + p_i + {p_i}^2 + \ldots + {p_i}^{ei})$

CSP2021-J1初赛

- k为 i × k 的最小质因子

- c[i × k] ＝ 最小质因子的指数
- f[i × k] ＝ 约数个数
- g[i × k] ＝ 约数和
- d[i × k] ＝ 辅助计算g

```
11 void init()
12 {
13     f[1] = g[1] = 1;
14     for (int i = 2; i <= n; i++) {
15         if (!a[i]) {
16             b[m++] = i;
17             c[i] = 1, f[i] = 2;
18             d[i] = 1, g[i] = i + 1;
19         }
20         for (int j = 0; j < m && b[j] * i <= n; j++) {
21             int k = b[j];
22             a[i * k] = 1;
23             if (i % k == 0) {
24                 c[i * k] = c[i] + 1;
25                 f[i * k] = f[i] / c[i * k] * (c[i * k] + 1);
26                 d[i * k] = d[i];
27                 g[i * k] = g[i] * k + d[i];
28                 break;
29             }
30             else {
31                 c[i * k] = 1;
32                 f[i * k] = 2 * f[i];
33                 d[i * k] = g[i];
34                 g[i * k] = g[i] * (k + 1);
35             }
36         }
37     }
38 }
```

质因子与约数

	质因子	约数
数量	$O(\log n)$，实际更小	$O(\sqrt{n})$
算法	2到√n试除，每次除尽	2到√n试除，每次找一对
时间复杂度	$O(\sqrt{n})$	$O(\sqrt{n})$
空间复杂度	$O(\log n)$	$d(n)$近似$O(\sqrt[3]{n})$
跳跳法 求区间内所有数的信息	$O(n\log\log n)$	$O(n\log n)$

倍数与约数

- 给定序列 a[1~n]，值域 V=[1,10^6]
- 对于值域中所有 x：
  - 求cnt1[x] = x 的倍数数量
  - 求cnt2[x] = x 的约数数量

倍数与约数

- 给定序列 a[1~n]
- 值域 V=[1,10^6]
  - cnt1[x] ＝x的倍数数量
  - cnt2[x] ＝x的约数数量

- 跳跳法O(VlogV)
- 高维前缀/后缀和
- O(VloglogV)

```
//cntp=值域[1,V]中质数数量
void work1(){//求cnt1
    for(int j=1;j<=cntp;++j){
        int lim = V/prime[j];
        for(int k=lim;k>=1;--k){
            cnt[k] += cnt[k*prime[j]];
        }
    }
}

void work2(){//求cnt2
    for(int j=1;j<=cntp;++j){
        int lim = V/prime[j];
        for(int k=1;k<=lim;++k){
            cnt[k*prime[j]] += cnt[k];
        }
    }
}
```

最大公约数gcd

- 设 $d = \gcd(x,y)$，那么d的唯一分解中任意质数的指数e_i为x,y的唯一分解中此质数指数的min

- 例如：$\gcd(180,240) = 60$
  - $180 = 2^2 \cdot 3^2 \cdot 5^1$
  - $240 = 2^4 \cdot 3^1 \cdot 5^1$
  - $60 = 2^2 \cdot 3^1 \cdot 5^1$

- 求gcd(x,y)的一种方法：将x,y分别质因数分解，每项指数取min，复杂度$O(\sqrt{x}+\sqrt{y})$

最小公倍数lcm

- 若$x|m$且$y|m$，称为m为x和y的公倍数
- 公倍数中最小的称为最小公倍数$\mathrm{lcm}(x,y)$

- $\mathrm{lcm}(x,y)$的唯一分解中任意质数的指数e_i为x,y的唯一分解中此质数指数的max
- 求$\mathrm{lcm}(x,y)$的一种方法：将x,y分别质因数分解，每项指数取max，复杂度$O(\sqrt{x}+\sqrt{y})$

- 先除后乘：$\mathrm{lcm}(x,y) = x/\_\mathrm{gcd}(x,y) * y$

P1029 最大公约数和最小公倍数问题

- 输入二个正整数$x, y(2 \leq x, y \leq 1000000)$，求出满足下列条件的P,Q的个数
  - 1. P,Q是正整数
  - 2. 要求P,Q以x为最大公约数，以y为最小公倍数

- 暴力枚举y约数，然后检查
- 复杂度O(NlogN)，是可以过的

```
for(int i=1;i*i<=y;++i){
    if(y%i) continue;
    fac.push_back(i);
    if(i != y/i) fac.push_back(y/i);
}

int ans = 0;
for(int i:fac){
    for(int j:fac){
        int d = __gcd(i,j);
        if(d==x && (ll)i*j/d==y){
            ++ans;
        }
    }
}
cout<<ans;
```

- 将x, y分解质因数，对于质因子p_i，若x, y中分解出的指数不相等，则有2种可能，否则有一种可能。

- 算法2
- 将x, y分解质因数，判断每个质因子的指数即可
  - e[x] < e[y]，方案数加倍
  - e[x] = e[y]，方案数不变
  - e[x] > e[y]，无解

- 算法3
- 将 y/x 分解质因数，设分解出的质因子个数为t，答案为2^t

P1072 [NOIP2009T2] Hankson的趣味题

- 已知正整数a0, a1, b0, b1, 正整数x满足：
  - 1. x 和 a0 的最大公约数是 a1；
  - 2. x 和 b0 的最小公倍数是 b1。
- 求满足条件的x的个数
  - 50%的数据，有 $1 \le a0, a1, b0, b1 \le 10000$ 且 $n \le 100$
  - 100%的数据，有 $1 \le a0, a1, b0, b1 \le 2{,}000{,}000{,}000$ 且 $n \le 2000$。

- **算法1：遍历[a1, a0]枚举x，然后计算gcd和lcm，50分**
- **算法2：将a0, a1, b0, b1分解质因数，乘法原理：考虑对每个质因数P，实际上限定了2个区间（或退化成单点），方案数贡献为它们交集的大小**
  - $\min(x, a0) = a1$
  - $\max(x, b0) = b1$

唯一分解-总结

- 唯一分解的结构
- 约数的结构
- 找一个数的质因子和约数
- gcd/lcm实际上是唯一分解上的指数min/max

拓展GCD

Gcd递归定理（欧几里得算法）

- $\gcd(a, b) = \gcd(b, a\%b)$
  - 设 $d = \gcd(a, b)$，有d整除x, b的线性组合（b和a%b），则d也是(b, a%b)的公约数，那么有：
    - $\gcd(a, b) \leq \gcd(b, a\%b)$
  - 同理可证 $\gcd(a, b) \geq \gcd(b, a\%b)$
  - (a, b)和(b, a%b)的公约数集合相同

欧几里得算法

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

```
ll gcd(ll a, ll b){
    return (b==0) ? a : gcd(b, a%b);
}
```

- 复杂度O(loga)
  - **1)若**$b \le a/2$，$a' \le a/2$
  - **2)若**$b > a/2$，$a'' = b' = a - b < a/2$

拓展欧几里得

- $d = gcd(a, b)$是a, b线性组合集合中的最小正元素[裴蜀定理]
- 不妨设 $d = gcd(a, b) = ax + by$，试求三元组(d, x, y)。可以理解为不定方程 $ax + by = d$ 的整数解，或者直线 $ax + by = d$ 上的整点坐标
- 首先计算出满足

$$d' = gcd(b, a\%b) = bx' + (a\%b)y'$$

- 的一组(d', x', y')，则有：

$$d = d' = bx' + (a - (a/b)b)y' = ay' + b(x' - (a/b)y')$$

- 那么可得一组满足要求的(d, x, y)为：
  - $d = d'$
  - $x = y'$
  - $y = x' - (a/b)y'$
- 边界条件如果$b = 0$，方程变为：$d = ax$，此时直接返回三元组$(a, 1, 0)$
- 复杂度$O(\log a)$，和欧几里得算法一样

拓展欧几里得（代码）

```
ll d,x,y;
void exgcd(ll a, ll b){
    if(b==0){
        x = 1;
        y = 0;
        d = a;
    }
    else{
        exgcd(b, a%b);
        ll t = x;
        x = y;
        y = t - a/b*y;
    }
}
//ExEuclid(a,b)执行完之后，全局变量(d,x,y)即为一组解
```

通解为：

$$x = x0 + k \times (b/d)$$

$$y = y0 - k \times (a/d)$$

其中$k \in Z$

U87176 zjl的兔子

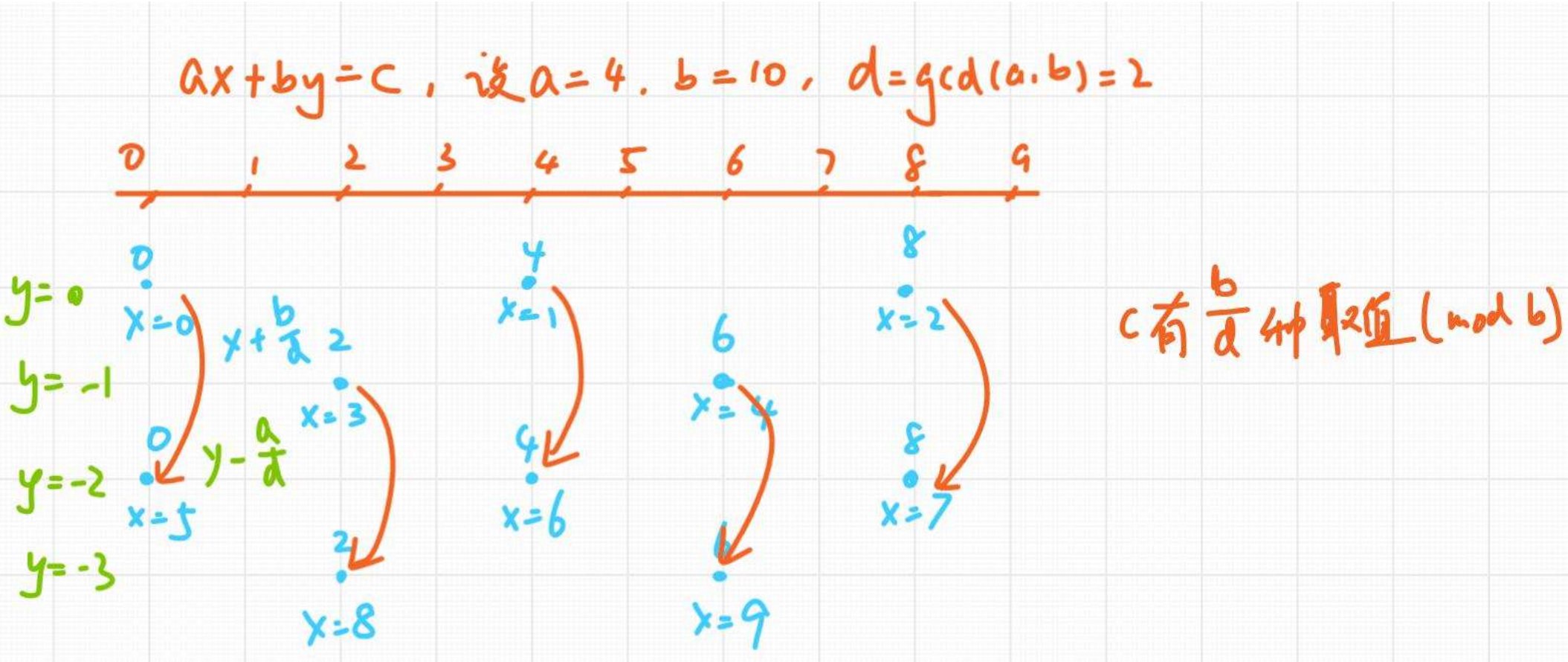
zjl的兔子在一个长度 n 米的圆形跑道（包含 $x = 0, 1, ..., n-1$ 这 n 个点）上，兔子从起点 $x = s$ 开始跳，每一步向前或者向后跳 y 米，并且兔子可以不限步数地在跑道上跳。

跑道上的 n 个点中，有些点兔子能够跳到，有些点无论如何兔子也跳不到。

例如 $n = 5$，兔子从 $x = 0$ 出发，$y = 2$，那么：$0 \to 2 \to 4 \to 1 \to 3$，兔子可以跳到 5 个点

例如 $n = 6$，兔子从 $x = 1$ 出发，$y = 3$，那么：$1 \to 4$，兔子只能跳到 2 个点

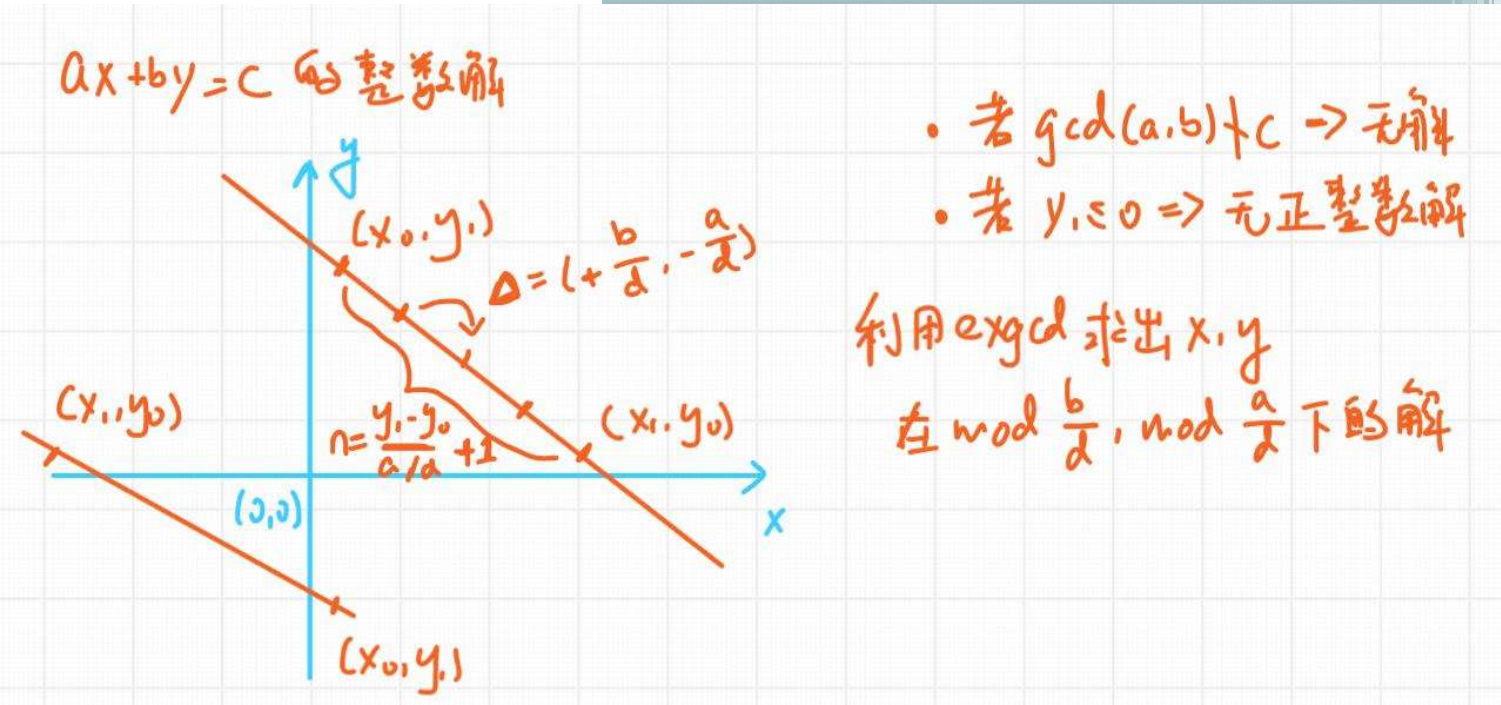
现在给定 n, s, y，请问兔子能跳到的点有多少个。


ax+by=c，设a=4，b=10，d=gcd(a,b)=2
0 1 2 3 4 5 6 7 8 9
y=0
y=-1
y=-2
y=-3
x+b/d
y-a/d
x=0
x=5
x=3
x=8
x=1
x=6
x=4
x=9
x=2
x=7
c有b/d种取值(mod b)

P5656 【模板】二元一次不定方程

- 给定整数a, b, c，对于二元一次不定方程：
  - ax+by=c

- 若该方程无整数解，输出 -1
- 若该方程有整数解，且有正整数解，则输出其正整数解的数量，所有正整数解中 x 的最小值，所有正整数解中 y 的最小值，所有正整数解中 x 的最大值，以及所有正整数解中 y 的最大值；
- 若方程有整数解，但没有正整数解，你需要输出所有整数解中 x 的最小正整数值，y 的最小正整数值



- **执行exgcd(a,b)之后**
  - x *= c/d
  - y *= c/d

- x**在**mod(b/d)**意义下有解**
- y**在**mod(a/d)**意义下有解**

- **特解（模意义下的最小解，注意特判0）：**
  - x0 = (x%(b/d) + (b/d))%(b/d);
  - y0 = (y%(a/d) + (a/d))%(a/d);

[NOIP2012T1]同余方程

- 求关于 x 的同余方程 ax = 1(mod p) 的最小正整数解。输入数据保证一定有解（意味着什么？）

- 存在逆元当且仅当 (a,p)互质
- 解不定方程 ax + py = 1即可，求出 x 在(mod p)下的解

逆元

- 在%p意义下：$\mathbf{a/b} = \mathbf{a}b^{-1} \pmod p$
- 若$\gcd(x, p) = 1$，则：
  - **exgcd**：$ax + py = 1$ 在 $\bmod p$ 意义下的解
- 若$\gcd(b, p)! = 1$，不存在逆元，此时做除法只能分解质因数了
- 若p为质数，则 $b^{-1} = b^{p-2} \pmod p$ [费马小定理]
- 可以在O(nlogn)/O(n)时间内处理1~n逆元/阶乘逆元
  - $n!^{-1} = \frac{(n+1)}{(n+1)!}$
  - $n^{-1} = \frac{(n-1)!}{n!}$
  - 式子中的阶乘实际上是前缀积，可以解决一般情况

欧拉定理

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

- 要求a与p互质。
- $\varphi(x)$为欧拉函数，定义为小于等于x且与x互质的数的个数。
  - 证明方法：$(0, \mathrm{p})$中与p互质的$\varphi(p)$个数构成模p乘法群

费马小定理

$$a^{p-1} \equiv 1 \pmod{p}$$

- 要求p为质数，且a不是p的倍数。
- 特别地，a可以为0。

威尔逊定理

对于素数 p 有 $(p-1)! \equiv -1 \pmod{p}$.

我们知道 $\mathbf{Z}_p$ 中所有非零元素 a 都有逆元 a^{-1}，于是 $\mathbf{Z}_p$ 中彼此互逆的元素乘积为 $\overline{1}$.

但是要注意 a 和 a^{-1} 可能相等。若 $a = a^{-1}$，则 $a^2 \equiv 1 \pmod{p}$，即

$$0 \equiv a^2 - 1 \equiv (a+1)(a-1) \pmod{p}$$

从而 $a \equiv 1 \pmod{p}$ 或 $a \equiv -1 \pmod{p}$.

这说明 $\mathbf{Z}_p \setminus \{\overline{0}, \overline{1}, \overline{-1}\}$ 中所有元素的乘积为 $\overline{1}$，进而 $\mathbf{Z}_p$ 中所有非零元素的积为 $\overline{-1}$.

P1516 青蛙的约会

- 2只青蛙在长度为L的环形数轴（纬度线）上，青蛙A,B的起点分别为x,y，每次跳跃长度为m,n，请问跳几次之后才能碰面。
- $x, y, m, n, L < \text{int}$范围

- $(x + km)\%L = (y + kn)\%L$ 的最小非负整数解k
- 解不定方程 $(m - n)k + Lt = x - y$

中国剩余定理

中国剩余定理 (Chinese Remainder Theorem, CRT) 可求解如下形式的一元线性同余方程组（其中 $n_1, n_2, \cdots, n_k$ 两两互质）：

$$\begin{cases} x & \equiv a_1 \pmod{n_1} \\ x & \equiv a_2 \pmod{n_2} \\ & \vdots \\ x & \equiv a_k \pmod{n_k} \end{cases}$$

中国剩余定理

1. 计算所有模数的积 n;

2. 对于第 i 个方程:

   a. 计算 $m_i = \frac{n}{n_i}$;

   b. 计算 m_i 在模 n_i 意义下的 逆元 m_i^{-1};

   c. 计算 $c_i = m_i m_i^{-1}$（**不要对 n_i 取模**）。

3. 方程组在模 n 意义下的唯一解为：$x = \sum_{i=1}^{k} a_i c_i \pmod{n}$。

扩展：模数不互质的情况

两个方程

设两个方程分别是 $x \equiv a_1 \pmod{m_1}$、$x \equiv a_2 \pmod{m_2}$；

将它们转化为不定方程：$x = m_1 p + a_1 = m_2 q + a_2$，其中 p, q 是整数，则有 $m_1 p - m_2 q = a_2 - a_1$。

由 裴蜀定理，当 $a_2 - a_1$ 不能被 $\gcd(m_1, m_2)$ 整除时，无解；

其他情况下，可以通过 扩展欧几里得算法 解出来一组可行解 (p, q)；

则原来的两方程组成的模方程组的解为 $x \equiv b \pmod{M}$，其中 $b = m_1 p + a_1$，$M = \operatorname{lcm}(m_1, m_2)$。

P4777 【模板】扩展中国剩余定理（EXCRT）

```
ll excrt(){
    ll ans = 0, P = 1;
    //ans + x*P = a[i] (mod p[i])
    //x*P = a[i] - ans (mod p[i])
    for(int i=1;i<=n;i++){
        exgcd(P, p[i]);
        ll c = a[i] - ans;
        if(c%d) return -1;
        x *= c/d;
        ll mod = p[i]/d;
        x = (x%mod + mod) % mod;
        ll Pt = P;
        P = lcm(P, p[i]);
        ans = (ans + x * Pt) % P;
    }
    return ans;
}
```

拓展GCD – 总结

- ax+by=c 解的结构，zjl的兔子模型
- 逆元
- 欧拉定理，费马小定理，威尔逊定理
- 中国剩余定理

P8807 [蓝桥杯 2022 国 C] 取模

- 给定 $n, m \sim 10^9$，判定是否存在 $1 \le x < y \le m$ 且 $n \pmod x = n \pmod y$

- 如果答案为NO，则对于$i = 1 \sim m$，$n \pmod i$ 各不相同
  - 对于$i = 1 \sim m, n = i - 1 \pmod i$
  - CRT：答案在 $\bmod L = \operatorname{lcm}(1,2,\ldots,m)$ 下唯一
  - $L - 1$符合条件
  - 答案为 NO 当且仅当 $n = L - 1 \pmod L$

CF2013E Prefix GCD

- 给一个序列
  a[1~n]，对其进行重排，最小化所有前缀的gcd之和：
  - gcd(a[1]) + gcd(a[1~2]) + … + gcd(a[1~n])
- n, a[i] = 100000

- 贪心：最小值放开头，每次选能让gcd减小最多的
- 最大化怎么做？
- 沿着前缀gcd阶梯进行DP：
  - cnt[x] = x倍数数量
  - dp[x] = (阶梯末尾高度)当前gcd = x，阶梯最大面积
  - 枚举阶梯上一层高度y(x|y)转移：
    - dp[x] = max(dp[y] + (cnt[x] − cnt[y]) ∗ x)
  - ans = dp[1]

除法分块

除法分块

- [CQOI2007]余数之和sum
  - 给出正整数n和k，计算 $\sum_{p=1}^{n} k\%p$ 的值
  - $1 \le n, k \le 1e9$

- $\sum_{p=1}^{n} k\%p = \sum_{p=1}^{n} (k - \frac{k}{p} * p) = nk - \sum_{p=1}^{n} \frac{k}{p} * p$
  - k/p只有√n个值
  - 块[l,r]内设c=k/l，则c$\sum_{p=l}^{r} p$的值可O(1)获得（等差数列求和公式，更一般情况可以O(n)预处理p的前缀和）
  - 注意当p>k时k/l=0,此时直接break即可
  - 总复杂度O(√n)

```
for(ll l=1,r; l<=N ; l=r+1) {
    if(K/l==0) break;
    r = min(K/(K/l), N);
    ans -= (K/l)*sum(l,r);
}
```