数论专题

同余

河南省实验中学信息技术组

2026 年 02 月 03 日

数论专题

- 质数
- 约数
- 同余

同余

- 若整数 a 和 b 除以正整数 m 的余数相等，则称 a, b 模 m 同余，记为 $a \equiv b (\mod m)$。
- 同余类：对于 $\forall a \in [0, m-1]$，集合 $\{a + km\}$ 的所有数模 m 同余，余数都是 a。该集合称为一个模 m 的同余类，记为 $\overline{a}$。
- 剩余系：模 m 的同余类一共有 m 个，分别为 $\overline{0}, \overline{1}, \overline{2}, \cdots, \overline{m-1}$，它们构成 m 的完全剩余系。
- 简化剩余系：$1 \sim m$ 中与 m 互质的数代表的同余类有 $\varphi(m)$ 个，它们构成 m 的简化剩余系。例如，模 10 的简化剩余系为 $\overline{1}, \overline{3}, \overline{7}, \overline{9}$。

同余相关定理

- 费马小定理：若 p 是质数，则对于任意的 a，有 $a^p \equiv a(\mod p)$。
- 欧拉定理：若正整数 a, n 互质，则 $a^{\varphi(n)} \equiv 1(\mod n)$，$\varphi(n)$ 为欧拉函数。
- 欧拉定理推论：若正整数 a, n 互质，则对于任意的正整数 b，有 $a^b \equiv a^{b \bmod \varphi(n)}(\mod n)$。特别地，正整数 a, n 不一定互质且 $b > \varphi(n)$ 时，有有 $a^b \equiv a^{b \bmod \varphi(n)+\varphi(n)}(\mod n)$。
- 小应用：很多题目要求答案模大质数 p 输出，对于 a^b 可以先对 a 模 p 然后对 b 模 $\varphi(p)$，再计算乘方，即 $a^b \equiv (a \mod p)^{b \bmod \varphi(p)}(\mod p)$。

【例】最幸运的数

【题目描述】
给定一个正整数 L，问至少多少个 8 连在一起组成的正整数是 L 的倍数?
【输入格式】
多行，每行包含一个正整数 $L(L \leq 2 \times 10^9)$。
【输出格式】
多行，对于每一个正整数 L，多少个 8 连在一起是 L 的倍数。

【样例输入】

```
8
11
16
```

【样例输出】

```
1
2
0
```

【样例解释】

- 第一组数据，1 个 8 就是 8 的倍数。
- 第二组数据，2 个 8 为 88 是 11 的倍数。
- 第三组数据，找不到解。

【例】最幸运的数

- x 个 8 连在一起组成的正整数可以写为 $\frac{8(10^x-1)}{9}$，那么我们要找到一个最小的 x，使得 $L|\frac{8(10^x-1)}{9}$。

$$L|\frac{8(10^x-1)}{9} \Leftrightarrow 9L|8(10^x-1) \Leftrightarrow \frac{9L}{\gcd(9L,8)}|\frac{8}{\gcd(9L,8)}(10^x-1) \Leftrightarrow$$

$$\frac{9L}{\gcd(9L,8)}|(10^x-1) \Leftrightarrow 10^x \equiv 1(\mod \frac{9L}{\gcd(9L,8)})$$

- 引理：若正整数 a,n 互质，则满足 $a^x \equiv 1(\mod n)$ 的最小正整数 x_0 是 $\varphi(n)$ 的约数。
- 若 10 和 $\frac{9L}{\gcd(9L,8)}$ 不互质，则 L 一定时 10 的倍数，肯定无解。
- 求出欧拉函数 $\varphi(\frac{9L}{\gcd(9L,8)})$，枚举它的所有约数，用快速幂逐一检查是否满足条件。
- 时间复杂度：$O(\sqrt{L}\log L)$

【例】最幸运的数

```
long long M = 9 * L / gcd(9 * L, 8);
long long n = phi(M);
if(gcd(10, M) != 1) { puts("0"); continue; }   // 不互质 无解
long long ans = n + 1;
for(long long i = 1; i * i <= n; ++i)
    if(n % i == 0)
    {
        if(fastpow(10, i, M) == 1) { ans = min(ans, i); break; }
        if(i != n / i) if(fastpow(10, n / i, M) == 1) ans = min(ans, n / i);
    }
if(ans == n + 1) puts("1"); else printf("%lld\n", ans);
```

【例】小凯的疑惑

【题目描述】

给你两个数 a, b，且它们互质，求 a, b 不能表示的最大的数。

【输入格式】

一行，两个数 $a, b (1 \leq a, b \leq 10^9)$。

【输出格式】

一行，a, b 不能表示的最大数。

【样例输入】

```
3 7
```

【样例输出】

```
11
```

【样例解释】

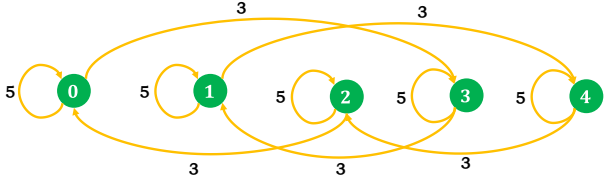
用 3 和 7 不能表示的数有 $1, 2, 4, 5, 8, 11$，其中最大的为 11。

【例】小凯的疑惑

- 给你两个数 $a, b(1 \leq a, b \leq 10^9)$，且它们互质，求 $ax + by(x \geq 0, y \geq 0)$ 不能表示的最大数。
- 方法 1：如果不考虑 $x \geq 0, y \geq 0$，因为 $\gcd(a, b) = 1$，$ax + by = m(m \in \mathbf{Z})$ 一定有解。
- 所以若出现不能表示的情况，则说明 $x \geq 0, y < 0$ 或 $x < 0, y \geq 0$。
- 首先讨论 $x \geq 0, y < 0$ 的情况，若使 $ax + by$ 尽量大，则 $y = -1$，
  - 当 $x \geq b$ 时，$ax - b = ab + a(x - b) - b = (a - 1)b + a(x - b)$，此时一定可以被 a, b 表示；
  - 当 $x < b$ 时，我们取 $x = b - 1$，得到 $ax - b = a(b - 1) - b = ab - a - b$，不能被 a, b 表示。
- 同理可证明或 $x < 0, y \geq 0$ 的情况，综上答案为 $ab - a - b$。

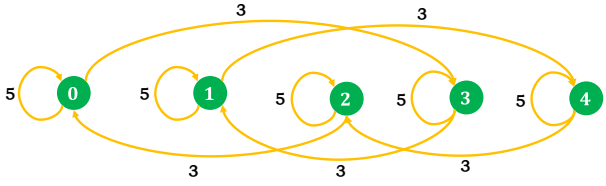
【例】小凯的疑惑

- 方法 2：同余类 BFS
- 首先模 a 的同余类有 $\overline{0}, \overline{1}, \overline{2}, \cdots, \overline{a-1}$，用 $d[x]$ 来表示同余类 $\overline{x}$ 最小能被 a, b 表示的数。
- 此时，以模 a 的同余类构建 a 个点，从每个点 x 到其本身 $(x+a)\%a$ 连一条长为 a 的路径(自环)，然后从每个点 x 到 $(x+b)\%a$ 连一条长度为 b 的路径。



- 此时，从 0 点开始在图中随机游走的产生的路径长度为 $k_1a + k_2b$ $(k_1, k_2 \in \mathbf{N})$，也就是能表示所有整数。

【例】小凯的疑惑



- 其中从 0 点走长度为 b 的边到达的点有 $b\%a, 2b\%a, 3b\%a, \cdots, (a-1)b\%a, 0$，这 a 个值一定互不相同。
- 所以 $b\%a, 2b\%a, 3b\%a, \cdots, (a-1)b\%a, 0$ 遍历了 a 的同余类各一次。
- 所以 d 数组中的值一定是 $\{b\%a, 2b\%a, 3b\%a, \cdots, (a-1)b\%a, 0\}$(顺序不定)，即各个同余类可以表示的最小值 (同余类图中从 0 点到各点的最短路)。
- 那么最大不能表示的数为 $\{b-a, 2b-a, 3b-a, \cdots, (a-1)b-a, 0-a\}$。
- 答案为 $ab-a-b$。

扩展欧几里得算法

- 裴蜀定理 (Bézout 定理)：对于任意的整数 a, b，存在一对整数 x, y，满足 $ax + by = \gcd(a, b)$。
- 用欧几里得算法证明 (数学归纳法)：
  - 当 $b = 0$ 时，显然有一对整数 $x = 1, y = 0$。
  - 当 $b > 0$ 时，有 $\gcd(a, b) = \gcd(b, a \mod b)$。假设存在一对整数 x, y，满足 $b \times x + (a \mod b) \times y = \gcd(b, a \mod b) = \gcd(a, b)$，因为 $b \times x + (a \mod b) \times y = b \times x + (a - b \times \lfloor \frac{a}{b} \rfloor) \times y = a \times y + b \times (x - \lfloor \frac{a}{b} \rfloor \times y)$，令 $x' = y, y' = x - \lfloor \frac{a}{b} \rfloor \times y$，就得到了 $ax' + by' = \gcd(a, b)$。
- 裴蜀定理是用欧几里得算法证明的，并且在证明的过程中给出了求解方法，这种方法被称为扩展欧几里得算法。

扩展欧几里得算法

- 扩展欧几里得算法得出方程的一组特解 x_0, y_0，并返回 a, b 的最大公约数 $d = \gcd(a, b)$。
- 对于更一般的方程 $ax + by = m$，它有解当前仅当 $d|m$，它的特解为 $x_0 = \frac{m}{d}x_0, y_0 = \frac{m}{d}y_0$。它的通解可以表示为：

$$x = \frac{b}{d}k + x_0, y = -\frac{a}{d}k + y_0 (k \in (Z)$$

最小正整数解为 $x_0 \mod \frac{b}{d}, y_0 \mod \frac{a}{d}$。

```
// x y 用引用 函数结束后需要其作为特解
int exgcd(int a, int b, int &x, int &y)
{
    if(b == 0) {x = 1, y = 0; return a;}
    // b * x + (a % b) * y = gcd(b, a % b)
    int d = exgcd(b, a % b, x, y);
    // x' = y y' = x - (a / b) * y;
    int z = x; x = y; y = z - (a / b) * y;
    return d;
}
```

【例】青蛙的约会

【题目描述】

两只青蛙在网上相识了，它们聊得很开心，于是觉得很有必要见一面。它们很高兴地发现它们住在同一条纬度线上，于是它们约定各自朝西跳，直到碰面为止。可是它们出发之前忘记了一件很重要的事情，既没有问清楚对方的特征，也没有约定见面的具体位置。不过青蛙们都是很乐观的，它们觉得只要一直朝着某个方向跳下去，总能碰到对方的。但是除非这两只青蛙在同一时间跳到同一点上，不然是永远都不可能碰面的。为了帮助这两只乐观的青蛙，你被要求写一个程序来判断这两只青蛙是否能够碰面，会在什么时候碰面。

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。

【例】青蛙的约会

【输入格式】

一行 5 个整数 $x, y, m, n, L (x, y, m, n, L \leq 2.1 \times 10^9, x \neq y)$。

【输出格式】

一行一个整数，表示碰面所需要的跳跃次数。

如果永远不可能碰面，则输出 `Impossible`。

【样例输入】

```
1 2 3 4 5
```

【样例输出】

```
4
```

【例】青蛙的约会

- 设青蛙跳了 k_1 步，那么青蛙 A 到达 $x+mk_1$，青蛙 B 到达 $y+nk_1$，因为纬线的总长度为 L，如果相遇，那么有

$$x+mk_1 \equiv y+nk_1 (\mod L)$$

$$x+k_1*m = y+k_1*n-k_2*L$$

$$(m-n)k_1+L*k_2 = y-x$$

- 令 $a=m-n, b=L, c=y-x$，那么方程变为 $ak_1+bk_2=c$，那么：
  1. 如果 c 不是 $\gcd(a,b)$ 的倍数，那么无解。
  2. 否则，令 $d=\gcd(a,b)$，先利用扩展欧几里得算法求出 $ak_1+bk_2=\gcd(a,b)$ 的特解 k_1'，那么方程 $ak_1+bk_2=c$ 的特解为 $k_1'=\frac{c}{d}k_1'$，那么方程的通解为 $k_1'=k_1'+k\frac{b}{d}$ 最小解为 $k_1' \mod \frac{b}{d}$。

【例】青蛙的约会


```cpp
// x+km===y+kn(mod L)
// x+k1*m=y+k1*n-k2*L
// (m-n)k1+L*k2=y-x
long long a = (m - n), b = L, c = y - x;
if(a < 0) a = -a, c = -c;
long long k1, k2;
long long d = exgcd(a, b, k1, k2);
if(c % d) { puts("Impossible"); return 0; }
k1 = k1 * c / d;
long long M = b / d;
k1 = (k1 % M + M) % M;
cout << k1;
```

乘法逆元

- 若整数 b, m 互质，则存在一个整数 x，使得 $bx \equiv 1 (\mod m)$，则称 x 为 b 的模 m 的乘法逆元，记为 $b^{-1} (\mod m)$。
  - 若 m 是质数并且 $b < m$，根据欧拉定理，$b^{\varphi(m)} \equiv b^{m-1} \equiv 1 (\mod m)$，即 $b \times b^{m-2} \equiv 1 (\mod m)$。因此，当模数 m 是质数时，b^{m-2} 即为 b 的乘法逆元。
  - 若 m 不是质数，那么乘法逆元可以通过求解线性同余方程 $bx \equiv 1 (\mod m)$ 的到。
- 小应用：若整数 b, m 互质，并且 $b|a$，则 $a/b \equiv a \times b^{-1} (\mod m) (\mod m)$。所以，在计算时遇到 a/b 对大质数 P 取余时，等价于计算 $a \times b^{-1} (\mod P) \mod P$，即

$$a/b \equiv a \times b^{-1} (\mod P) (\mod P) \equiv a \times b^{P-2} (\mod P)$$

【例】Sumdiv

【题目描述】
求 A^B 的所有约数之和，结果对 9901 取余。
【输入格式】
一行，两个数 $A, B(1 \leq A, B \leq 5 \times 10^7)$。
【输出格式】
一行，一个数，A^B 的所有约数之和，结果对 9901 取余。

【样例输入】

```
2 3
```

【样例输出】

```
15
```

【样例解释】
$2^3 = 8$，它的约数有 $1, 2, 4, 8$，总和为 15。

【例】Sumdiv

- 把 A 分解质因数，表示为 $p_1^{c_1}p_2^{c_2}\cdots p_m^{c_m}$，则 A^B 表示为 $p_1^{B\times c_1}p_2^{B\times c_2}\cdots p_m^{B\times c_m}$，它的所有约数和为

$$(1+p_1+p_1^2+\cdots+p_1^{B\times c_1})\times\cdots\times(1+p_m+p_m^2+\cdots+p_m^{B\times c_m})$$

$$=\frac{p^{B\times c_1+1}-1}{p_1-1}\times\cdots\times\frac{p^{B\times c_m+1}-1}{p_m-1}$$

- 用快速幂求 $(p^{B\times c_i+1}-1) \mod 9901$，并利用逆元求出 $(p_i-1) \mod 9901$。
  - 若 (p_i-1) 不是 9901 的倍数，即它们互质，那么求出 (p_i-1) 的逆元 $x=(p_i-1)^{9901-2}$，用乘 x 代替除法。
  - 若 (p_i-1) 是 9901 的倍数，此时乘法逆元不存在，但是 $p_i \mod 9901=1$，所以 $(1+p_i+p_i^2+\cdots+p_i^{B\times c_i})\equiv(1+1^2+1^3+\cdots+1^{B\times c_i})\equiv(B\times c_i+1)(\mod 9901)$。

【例】Sumdiv

```
int p[15], c[15];
int m = 0;
for(int i = 2; i * i <= a; ++i)
{
    if(a % i == 0) p[++m] = i, c[m] = 0;
    while(a % i == 0) ++c[m], a /= i;
}
if(a > 1) p[++m] = a, c[m] = 1;
long long ans = 1;
for(int i = 1; i <= m; ++i)
{
    if((p[i] - 1) % M == 0) ans = ans * (c[i] * b + 1) % M;
    else
    ans = ans * (fastpow(p[i], c[i] * b + 1) - 1 + M) % M * fastpow(p[i] - 1, M - 2) % M;
}
```

线性同余方程

- 给定整数 a, b, m，求一个整数 x 满足 $ax \equiv m(\mod b)$，或者给出无解。因未知数的指数是 1，所以程之为一次同余方程或线性同余方程。
- $ax \equiv m(\mod b) \Leftrightarrow b|(ax - m)$，设 $ax - m = b(-y)$，则原方程可以写为 $ax + by = m$。故，方程有解的条件是 $\gcd(a, b)|m$。
- 先用扩展欧几里得方法求出 $ax + by = \gcd(a, b)$ 的特解 x_0，则原方程的特解为 $x' = x_0 \frac{m}{d}$，其中 $d = \gcd(a, b)$。
- 方程的通解 $x = x' + k\frac{b}{d} = x_0 \frac{m}{d} + k\frac{b}{d} (k \in \mathbf{Z})$，即为所有模 $\frac{b}{d}$ 与 x' 同余的整数。

中国剩余定理[1]

设 $m_1, m_2, \cdots, m_n$ 是两两互质的整数，对于任意的 n 个正整数 $a_1, a_2, \cdots, a_n$，方程组

$$\begin{cases} x \equiv a_1 (\mod m_1) \\ x \equiv a_2 (\mod m_2) \\ \vdots \\ x \equiv a_n (\mod m_n) \end{cases}$$

有正整数解，解为 $x = \sum_{i=1}^{n} a_i M_i t_i$，方程的通解为 $x + kM (k \in \mathbf{Z})$。

其中 $M = \prod_{i=1}^{m} m_i, M_i = \frac{M}{m_i}$，$t_i$ 是线性方程 $M_i t_i \equiv 1 (\mod m_i)$ 的一个解(乘法逆元 $M_i^{m_i - 2}$)。

[1]有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二。问物几何?

【例】同余方程

【题目描述】

求关于 x 的同余方程 $ax \equiv 1(\mod b)$ 的最小正整数解。

【输入格式】

一行两个整数 $a, b(2 \leq a, b \leq 2 \times 10^9)$。

【输出格式】

一行一个整数 x_0，即最小正整数解。输入数据保证一定有解。

【样例输入】

```
3 10
```

【样例输出】

```
7
```

【例】同余方程

- 上述同余方程可以转化为 $ax + by = 1$，那么有解当且仅当 $\gcd(a, b) = 1$。
- 根据扩展欧几里得方法求出特解 x_0。
- 通解为所有模 $\frac{b}{\gcd(a,b)} = b$ 与 x_0 同余的整数。

【例】同余方程


```cpp
long long exgcd(long long a, long long  b, long long &x, long long & y)
{
    if(b == 0) {x = 1, y = 0; return a;}
    long long d = exgcd(b, a % b, x, y);
    long long z = x; x = y; y = z - y * (a / b);
    return d;
}

int main()
{
    long long a, b, x, y;
    cin >> a >> b;
    exgcd(a, b, x, y);
    cout << (x % b + b) % b << "\n";
```

【例】表达整数的奇怪方式

【题目描述】

给定 $2n$ 个正整数 $a_1, a_2, \cdots, a_n$ 和 $m_1, m_2, \cdots, m_n$，求一个最小的正整数 x，满足 $\forall i \in [1, n], x \equiv a_i (\bmod\ m_i)$，或者给出无解。

【输入格式】

第一行包含整数 $n(n \leq 25)$。

接下来 n 行，每行两个用空格隔开正整数 $m_i, a_i (0 \leq a_i < m_i \leq 2^{31} - 1)$。

【输出格式】

输出最小负整数 x，如果 x 不存在，则输出 -1。

如果存在 x，则保证 x 一定在 64 位整数范围内。

【样例输入】

```
2
8 7
11 9
```

【样例输出】

```
31
```

实际上本题要求解如下方程

$$\begin{cases} x \equiv a_1 (\mod m_1) \\ x \equiv a_2 (\mod m_2) \\ \vdots \\ x \equiv a_n (\mod m_n) \end{cases}$$

但是 $m_1, m_2, \cdots, m_n$ 不一定两两互质，所以不能直接使用中国剩余定理。

【例】表达整数的奇怪方式

- 首先考虑，如果只有两个方程

$$\begin{cases} x \equiv a_1 (\mod m_1) \\ x \equiv a_2 (\mod m_2) \end{cases} \Rightarrow \begin{cases} x = k_1 m_1 + a_1 \\ x = -k_2 m_2 + a_2 \end{cases} \Rightarrow k_1 m_1 + a_1 = -k_2 m_2 + a_2$$

- 那么经过移项合并后可得 $m_1 k_1 + m_2 k_2 = a_2 - a_1$，也即要找到合适的 k_1, k_2 使得等式成立 (要求 x 最小)。
- 利用扩展欧几里得算法，可以求解上述方程。
  - 如果 $a_2 - a_1$ 不是 $\gcd(m_1, m_2)$ 的倍数，则无解。
  - 否则通过扩展欧几里得算法可以求出方程 $m_1 k_1 + m_2 k_2 = \gcd(m_1, m_2)$ 的特解为 k_1', k_2'，那么原方程的特解 $k_1' = \frac{a_2 - a_1}{d} k_1', k_2' \frac{a_2 - a_1}{d} k_2'$ 那么可以得出通解为

$$\begin{cases} k_1 = k \frac{m_2}{d} + k_1' \\ k_2 = k \frac{m_1}{d} + k_2' \end{cases}$$

其中 $d = \gcd(m_1, m_2), k \in \mathbf{Z}$。

【例】表达整数的奇怪方式

- 将 k_1 带入第一个方程，则可以得到

$$x = k_1 m_1 + a_1 = (k_1' + k\frac{m_2}{d})m_1 + a_1 = k\frac{m_1 m_2}{d} + k_1' m_1 + a_1$$

其中，m_1, m_2, d, k_1', a_1 均为已知量。

- 上述方程可以写为

$$x \equiv (k_1' m_1 + a_1)(\mod \frac{m_1 m_2}{d})$$

将该方程与第三个方程联立可以得出一个新的方程组。

- 综上，经过 $n-1$ 次扩展欧几里得算法即可得出方程的解。

【例】表达整数的奇怪方式


```
long long a1, m1, a2, m2, a3, m3;
scanf("%lld%lld", &m1, &a1);
for(int i = 2; i <= n; ++i)
{
    scanf("%lld%lld", &m2, &a2);
    // x=k1*m1+a1=-k2*m2+a2
    // m1*k1+m2*k2=a2-a1
    long long k1, k2;
    long long d = exgcd(m1, m2, k1, k2);
    if(myabs(a2 - a1) % d) { puts("-1"); return 0; }
    k1 *= (a2 - a1) / d; // m1*k1+m2*k2=a2-a1 的解
    long long M = m2 / d;
    k1 = (k1 % M + M) % M; // 最小正整数解
    long long t = m1 * m2 / d;
    a1 = (k1 * m1 + a1) % t, m1 = t;   // 新方程
}
printf("%lld", (a1 % m1 + m1) % m1);
```

高次同余方程

- 给定正整数 a, b, m，其中 a, m 互质，求一个非负整数 x，使得 $a^x \equiv b \pmod{m}$。
- Baby Step,Giant Step 算法

练习

- 最幸运的数 (COGS 3435)
- 小凯的疑惑 [NOIP 2017](COGS 2864)
- 青蛙的约会 (COGS 1677)
- Sumdiv(COGS 2691)
- 魔法部落 (COGS 3264)
- 同余方程 [NOIP 2012](COGS 1265)
- 表达整数的奇怪方式 (COGS 3479)
- 韩信点兵 (COGS 1786)
- 线性同余发生器 (COGS 3709)
- Xiao 9* 大战朱最学 (COGS 2625)
- 学姐的巧克力盒 (COGS 2511)
- 计算器 (COGS 3246)