第八章 同余式

> **线性同余式定理**：设a，c与m是整数，$m \geq 1$，且设$g = gcd(a, m)$。
> (a) 如果$g \nmid c$，则同余式$ax \equiv c \pmod{m}$没有解
> (b) 如果$g \mid c$，则同余式$ax \equiv c \pmod{m}$恰好有g个不同的解。要求这些解，首先求线性方程$au + mv = g$的一个解(u_0, v_0)。则$x_0 = \frac{cu_0}{g}$是$ax \equiv c \pmod{m}$的解，不同解的完全集为
>
> $$x \equiv x_0 + k \cdot \frac{m}{g} \pmod{m}，k = 0, 1, 2, \ldots, g-1$$

> **模p多项式根定理**： 设p为素数，
>
> $$f(x) = a_0x^d + a_1x^{d-1} + \ldots + a_d$$
>
> 是次数为$d \geq 1$的整系数多项式，且p不整除a_0，则同余式
>
> $$f(x) \equiv 0 \pmod{p}$$
>
> 最多有d个的不同解

> **习题**

- *8.1* 假设$a_1 \equiv b_1 \pmod{m}$与$a_2 \equiv b_2 \pmod{m}$。
  - *(a)* 验证$a_1 + a_2 \equiv b_1 + b_2 \pmod{m}$与$a_1 - a_2 \equiv b_1 - b_2 \pmod{m}$。
  - *(b)* 验证$a_1a_2 \equiv b_1b_2 \pmod{m}$。

    (a)因为$a_1 \equiv b_1 (mod\ m)$，所以$a_1 - b_1 = mx$，同理$a_2 - b_2 = my$
    $a_1 - b_1 + a_2 - b_2 = (a_1 + a_2) - (b_1 + b_2) = m(x+y)$，即$a_1 + a_2 \equiv b_1 + b_2 (mod\ m)$得证
    $a_1 - b_1 - (a_2 - b_2) = (a_1 - a_2) - (b_1 - b_2) = m(x-y)$，即$a_1 - a_2 \equiv b_1 - b_2 (mod\ m)$得证
    (b) $a_1a_2 - b_1b_2 = a_1a_2 - a_1b_2 + a_1b_2 - b_1b_2 = a_1(a_2 - b_2) + (a_1 - b_1)b_2 = a_1my + a_2mx$
    即$a_1a_2 \equiv b_1b_2 (mod\ m)$得证
- *8.2* 假设$ac \equiv bc \pmod{m}$和$gcd(c, m) = 1$。证明$a \equiv b \pmod{m}$。

  8.2因为$ac \equiv bc (mod\ m)$，说明$ac - bc$可以被m整除，又因为$gcd(c, m) = 1$，所以$a - b$可以被m整除，即$a \equiv b (mod\ m)$得证
- *8.3* 求下述同余式的所有不同解。
  - *(a)* $7x \equiv 3 \pmod{15}$
  - *(b)* $6x \equiv 5 \pmod{15}$
  - *(c)* $x^2 \equiv 1 \pmod{8}$
  - *(d)* $x^2 \equiv 2 \pmod{7}$

- *(e)* $x^2 \equiv 3 \pmod{7}$

```
tangxianning@MacBook-Air GoProject % go run test.go
解同余式 7x ≡ 3 (mod 15):
所有解: [9]
解同余式 6x ≡ 5 (mod 15):
错误: gcd(6, 15) = 3 不整除 c = 5
tangxianning@MacBook-Air GoProject % go run test.go
解同余式 f(X) ≡ 0 (mod 8):
所有解: [1 3 5 7]
tangxianning@MacBook-Air GoProject % go run test.go
解同余式 f(X) ≡ 0 (mod 7):
所有解: [3 4]
tangxianning@MacBook-Air GoProject % go run test.go
解同余式 f(X) ≡ 0 (mod 7):
同余式 f(X) ≡ 0 (mod 7)无解
```

- *8.4* 证明下述整除性试验结果。
  - *(a)* 数a被4整除当且仅当它的末尾两位数被4整除。
  - *(b)* 数a被8整除当且仅当它的末尾三位数被8整除。
  - *(c)* 数a被3整除当且仅当它的各位数字之和被3整除。
  - *(d)* 数a被9整除当且仅当它的各位数字之和被9整除。
  - *(e)* 数a被11整除当且仅当它的各位数字交错和被11整除。

(a) 令 $a=100b+c$，c即为a的最后两位数，因为$100b$可以被4整除，所以a只有在c能被4整除时才能被4整除，得证。

(b) 令 $a=1000b+c$，c即为a的最后三位数，因为$1000b$可以被4整除，所以a只有在c能被8整除时才能被8整除，得证。

(c) 令 $a=a_1+10a_2+10^2a_3+10^3a_4+\cdots$，考虑到 $10\equiv1 \pmod{3}$，所以 $a\equiv a_1+a_2+a_3+\cdots \pmod{3}$，所以$a$只有在它各位数字之和能被3整除时能被3整除。

(d) 令 $a=a_1+10a_2+10^2a_3+10^3a_4+\cdots$，考虑到 $10\equiv1 \pmod{9}$，所以 $a\equiv a_1+a_2+a_3+\cdots \pmod{9}$，所以$a$只有在它各位数字之和能被9整除时能被9整除。

(e) 令 $a=a_1+10a_2+10^2a_3+10^3a_4+\cdots$，考虑到 $10\equiv-1 \pmod{11}$，所以 $a\equiv a_1-a_2+a_3\cdots \pmod{11}$，所以$a$只有在它各位数字之和能被11整除时能被11整除。

- *8.5* 求下述线性同余式的所有不同解。
  - *(a)* $8x \equiv 6 \pmod{14}$
  - *(b)* $66x \equiv 100 \pmod{121}$

- (c) $21x \equiv 14 \pmod{91}$

```
tangxianning@MacBook-Air GoProject % go run test.go
解同余式 8x ≡ 6 (mod 14):
所有解: [6 13]
解同余式 66x ≡ 100 (mod 121):
错误: gcd(66, 121) = 11 不整除 c = 100
解同余式 21x ≡ 14 (mod 91):
所有解: [5 18 31 44 57 70 83]
```

- 8.6 确定下述同余式的不同解的个数。无需求出解。
  - (a) $72x \equiv 47 \pmod{200}$
  - (b) $4183x \equiv 5781 \pmod{15087}$
  - (c) $1537x \equiv 2863 \pmod{6731}$

```
tangxianning@MacBook-Air GoProject % go run test.go
解同余式 72x ≡ 47 (mod 200):
错误: gcd(72, 200) = 8 不整除 c = 47
解同余式 4183x ≡ 5781 (mod 15087):
所有解: [225 546 867 1188 1509 1830 2151 2472 2793 3114 3435 3756 4077 4398 4719 5040 5361 5682 6003 6324 6645 6966 7287 7608 7929 8
250 8571 8892 9213 9534 9855 10176 10497 10818 11139 11460 11781 12102 12423 12744 13065 13386 13707 14028 14349 14670 14991]
解同余式 1537x ≡ 2863 (mod 6731):
错误: gcd(1537, 6731) = 53 不整除 c = 2863
```

- 8.7 编写程序解同余式$ax \equiv c \pmod{m}$。(如果$gcd(a,m)$不整除c，则输出出错信息和$gcd(a,m)$的值。)测试程序，求出习题8.6中同余式的所有解。

```go
package main

import (
    "fmt"
)

func extendedGCD(a, m int) (gcd, x, y int) {
    if m == 0 {
        return a, 1, 0
    }
    gcd, x1, y1 := extendedGCD(m, a%m)
    x = y1
    y = x1 - (a/m)*y1
    return
}

func solveCongruence(a, c, m int) interface{} {
    gcd, _, _ := extendedGCD(a, m)

    if c%gcd != 0 {
        return fmt.Sprintf("错误: gcd(%d, %d) = %d 不整除 c = %d", a, m, gcd, 
c)
    }

    aPrime := a / gcd
    cPrime := c / gcd
    mPrime := m / gcd
```

```go
	_, xInv, _ := extendedGCD(aPrime, mPrime)
	xInv = (xInv % mPrime + mPrime) % mPrime

	x0 := (xInv * cPrime) % mPrime

	var solutions []int
	for i := 0; i < gcd; i++ {
		solutions = append(solutions, (x0 + i*mPrime) % m)
	}

	return solutions
}

func main() {
	congruences := []struct {
		a, c, m int
	}{
		{72, 47, 200},
		{4183, 5781, 15087},
		{1537, 2863, 6731},
	}

	for _, congruence := range congruences {
		fmt.Printf("解同余式 %dx ≡ %d (mod %d): \n", congruence.a,
congruence.c, congruence.m)
		result := solveCongruence(congruence.a, congruence.c, congruence.m)
		switch r := result.(type) {
		case string:
			fmt.Println(r)
		case []int:
			fmt.Printf("所有解: %v\n", r)
		}
	}
}
```

```
tangxianning@MacBook-Air GoProject % go run test.go
解同余式 72x ≡ 47 (mod 200):
错误: gcd(72, 200) = 8 不整除 c = 47
解同余式 4183x ≡ 5781 (mod 15087):
所有解: [225 546 867 1188 1509 1830 2151 2472 2793 3114 3435 3756 4077 4398 4719 5040 5361 5682 6003 6324 6645 6966 7287 7608 7929 8
250 8571 8892 9213 9534 9855 10176 10497 10818 11139 11460 11781 12102 12423 12744 13065 13386 13707 14028 14349 14670 14991]
解同余式 1537x ≡ 2863 (mod 6731):
错误: gcd(1537, 6731) = 53 不整除 c = 2863
```

- *8.8* 编写程序，输入正整数m和整系数多项式$f(X)$，输出同余式$f(X) \equiv 0 \pmod{m}$的所有解。(无需细想，只需让X分别取0，1，2，$\cdots$，$m-1$，看看哪些值是解。)取多项式

$$f(X) = X^{11} + 21X^7 - 8X^3 + 8$$

对下述每个m值

$$m \in \{130, 137, 144, 151, 158, 165, 172\}$$

通过解同余式$f(X) \equiv 0 \pmod{m}$来测试程序。

```go
package main

import (
   "fmt"
   "math/big")

func evaluatePolynomial(X, m int) int {
   X11 := big.NewInt(int64(X))
   X11.Exp(X11, big.NewInt(11), nil).Mod(X11, big.NewInt(int64(m)))

   X7 := big.NewInt(int64(X))
   X7.Exp(X7, big.NewInt(7), nil).Mod(X7, big.NewInt(int64(m)))

   X3 := big.NewInt(int64(X))
   X3.Exp(X3, big.NewInt(3), nil).Mod(X3, big.NewInt(int64(m)))

   fX := big.NewInt(0)
   fX.Add(fX, X11)
   fX.Add(fX, X7.Mul(big.NewInt(21), X7))
   fX.Sub(fX, X3.Mul(big.NewInt(8), X3))
   fX.Add(fX, big.NewInt(8))

   fX.Mod(fX, big.NewInt(int64(m)))

   return int(fX.Int64())
}

func main() {
   mValues := []int{130, 137, 144, 151, 158, 165, 172}
   for _, m := range mValues {
      fmt.Printf("解同余式 f(X) ≡ 0 (mod %d): \n", m)
      sign := 0
      var solutions []int
      for X := 0; X < m; X++ {
         if evaluatePolynomial(X, m) == 0 {
            solutions = append(solutions, X)
            sign++
         }
      }
      if sign == 0 {
         fmt.Printf("同余式 f(X) ≡ 0 (mod %d)无解\n", m)
      } else {
         fmt.Printf("所有解: %v\n", solutions)
      }
```

```
    }
}
```

```
tangxianning@MacBook-Air GoProject % go run test.go
解同余式 f(X) ≡ 0 (mod 130):
所有解: [2 47 67 112]
解同余式 f(X) ≡ 0 (mod 137):
所有解: [99 104]
解同余式 f(X) ≡ 0 (mod 144):
同余式 f(X) ≡ 0 (mod 144)无解
解同余式 f(X) ≡ 0 (mod 151):
所有解: [84 105]
解同余式 f(X) ≡ 0 (mod 158):
所有解: [36 115]
解同余式 f(X) ≡ 0 (mod 165):
所有解: [122 137 152]
解同余式 f(X) ≡ 0 (mod 172):
所有解: [74 160]
```

- *8.9*
  - *(a)* 同余式

    $$X^4+5X^3+4X^2-6X-4 \equiv 0 \pmod{11}，0 \leq X < 11$$

    有多少个解？有4个解吗？还是有少于4个解？

    ```
    tangxianning@MacBook-Air GoProject % go run test.go
    解同余式 f(X) ≡ 0 (mod 11):
    所有解: [1 9]
    ```

  - *(b)* 考察同余式$X^2-1 \equiv 0 \pmod{8}$，当$0 \leq X < 8$时它有几个解？

    ```
    tangxianning@MacBook-Air GoProject % go run test.go
    解同余式 f(X) ≡ 0 (mod 8):
    所有解: [1 3 5 7]
    ```

- *8.10* 设p，q为不同素数。同余式

  $$X^2-a^2 \equiv 0 \pmod{pq}$$

最多可能有多少个解?

8.10 根据中国剩余定理，求解模 pq 的同余式可以分解为两个模 p 和 q 的问题，即分别解 $x^2 \equiv a^2 \pmod p$ 和 $x^2 \equiv a^2 \pmod q$

同余式 $x^2 \equiv a^2 \pmod p$ 只有两个解，这些解是 $x \equiv \pm a \pmod p$ 除非 $a \equiv 0 \pmod p$，此时只有一个解。模 q 时同理。所以最多可能的解数为模 p 和模 q 的最多解数的乘积，即4个。