Teoria dos Números

Aritmética Modular

Prof. Edson Alves

Faculdade UnB Gama

Aritmética Modular

Relação de congruência

Definição

Sejam a, b e m números inteiros tais que $m > 1$. Dizemos que a é congruente a b módulo m se m divide $a - b$.

Notação: $a \equiv b \pmod{m}$

Definição equivalente de congruência

- Sejam a, b e m números inteiros tais que $m > 1$
- Pela Divisão de Euclides, $a = mq_1 + r_1$, com $0 \leq r_1 < m$, e $b = mq_2 + r_2$, com $0 \leq r_2 < m$
- Assim, $a - b = m(q_1 - q_2) + (r_1 - r_2)$
- Se a e b são congruentes módulo m, então a diferença $a - b$ é um múltiplo de m
- Da expressão anterior, então $r_1 - r_2$ é múltiplo de m
- Como ambos restos são menores do que m, então $r_1 - r_2 = 0$

Definição equivalente de congruência

- Por outro lado, suponha que $r_1 = r_2$
- Neste caso, $a - b = m(q_1 - q_2)$, o significa que m divide $a - b$
- Logo, a é congruente a b módulo m
- Dados estes dois fatos, a congruência pode ser definida também por meio dos restos da divisão por m

Definição alternativa de congruência

Sejam a, b e m inteiros tais que $m > 1$. Dizemos que a e b são congruentes módulo m se ambos deixam mesmo resto quando divididos por m.

Propriedades da relação de congruência

Sejam a, b, c, m inteiros tais que $m > 1$. Valem as seguintes propriedades da congruência:

1. **propriedade reflexiva**: $a \equiv a \pmod{m}$
2. **propriedade simétrica**: se $a \equiv b \pmod{m}$ então $b \equiv a \pmod{m}$
3. **propriedade transitiva**: se $a \equiv b \pmod{m}$ e $b \equiv c \pmod{m}$ então $a \equiv c \pmod{m}$

Relação de equivalência

- Para verificar a propriedade reflexiva, observe que, para qualquer a inteiros, vale que $a - a = 0 = 0 \times m$
- Se $a \equiv b \pmod{m}$, então $a - b = mk$, para algum k inteiro
- Assim $b - a = -(a - b) = -mk$, isto é, $b \equiv a \pmod{m}$, de modo que vale a propriedade simétrica
- Por fim, se $a \equiv b \pmod{m}$ e $b \equiv c \pmod{m}$, então $a - b = mk_1$ e $b - c = mk_2$
- Logo
$$a - c = (a - b) + (b - c) = mk_1 + mk_2 = m(k_1 - k_2)$$
- Estas propriedades mostram que a relação de congurência é, de fato, uma relação de equivalência

Classes de equivalência módulo m

Definição

Seja m um inteiro tal que $m > 1$. O conjunto

$$[a] = \{b \in \mathbb{Z} \mid a \equiv b \pmod{m}\}$$

é denominado **classe de equivalência de a módulo m**.

O inteiro a é denominado **representante** da classe $[a]$.

Partição dos inteiros

- Sendo uma relação de equivalência, as classes de equivalência induzidas pela relação de congruência particionam o conjunto dos números inteiros em m conjuntos disjuntos
- Observem que, se $a \equiv b \pmod{m}$, então $[a] = [b]$
- Assim, convencionaremos que o representante de cada classe de equivalência x será o inteiro positivo r tal que $r \equiv x \pmod{m}$ e $0 \leq r < m$
- Ou seja, r corresponde aos restos euclidianos da divisão por m
- Com esta convenção, temos que

$$[0] \cup [1] \cup [2] \cup \ldots \cup [m-1] = \mathbb{Z},$$

com $[i] \cap [j] = \emptyset$ se $i \neq j$, $0 \leq i, j < m$

Conjunto das classes de equivalência

Definição

Seja m um inteiro positivo $m > 1$. Então

$$\mathbb{Z}_m = \{[0], [1], [2], \ldots, [m-1]\}$$

é o conjunto das classes de equivalência de m.

Adição em Z_m

- É possível definir uma operação de adição em $\mathbb{Z}_m$
- Seja $+ : \mathbb{Z}_m \times \mathbb{Z}_m \to \mathbb{Z}_m$ tal que $[a] + [b] = [c]$ se, e somente se, $c \equiv a + b \pmod{m}$
- Na prática, quando não houver ambiguidade, os colchetes que caracterizam as classes de equivalência podem ser omitidos
- Observe que esta adição lembra, mas não é idêntica, a adição nos inteiros
- Por exemplo, em $\mathbb{Z}_7$, $[5] + [4] = [2]$
- A leitura desta expressão é "somar um número que deixa resto 5 quando dividido por 7 a um número que deixa resto 4 quando dividido por 7 resulta em um número que deixa resto 2 quando dividido por 7"

Grupo das classes de equivalência

Proposição

Seja m um inteiro positivo tal que $m > 1$. Então $(\mathbb{Z}_m, +)$ é um grupo abeliano.

Demonstração

- Para verificar que $(\mathbb{Z}_m, +)$ é um grupo, observe inicialmente que a adição é fechada em $\mathbb{Z}_m$
- Isto é, a soma de dois inteiros a e b é um inteiro c que pertence a uma das classes de $\mathbb{Z}_m$, pois estas particionam os inteiros
- Além disso, $[0]$ é o elemento neutro da adição
- A associatividade e a comutatividade são consequências diretas destas propriedades para os inteiros
- Por fim, para qualquer classe $[a]$, a classe $[m - a]$ será seu inverso aditivo, pois

$$[a] + [m - a] = [0]$$

Multiplicação em $\mathbb{Z}_m$

- Também é possível definir uma multiplicação em $\mathbb{Z}_m$
- Seja $\times : \mathbb{Z}_m \times \mathbb{Z}_m \to \mathbb{Z}_m$ tal que $[a] \times [b] = [c]$ se, e somente se, $c \equiv ab \pmod{m}$
- Por exemplo, em $\mathbb{Z}_9$, $[4] \times [8] = [5]$
- A depender do valor de m, é possível que $\mathbb{Z}_m$ tenha **divisores de zero**, isto é, dois elementos diferentes de $[0]$ tais que seu produto resulte em $[0]$
- Em $\mathbb{Z}_{12}$, temos $[2] \times [6] = [0]$ e $[4] \times [3] = [0]$
- Dada uma classe $[a] \subset \mathbb{Z}_m$, a existência de um inverso multiplicativo também dependerá do valor de m

Grupo multiplicativo $\mathbb{Z}_p$

Proposição

Seja m um inteiro tal que $m > 1$. Então $(\mathbb{Z}_m, \times)$ será um grupo abeliano se, e somente se, m é um número primo.

Demonstração

- O fechamento, a associatividade e a comutatividade decorrem diretamente da multiplicação nos inteiros e da partição dos inteiros pelas classes de equivalência, e independem do valor de m
- Também, para qualquer m, a classe $[1]$ será o elemento identidade da multiplicação, isto é, para qualquer $[a] \subset \mathbb{Z}_m$,
$$[1] \times [a] = [a] \times [1] = [a]$$
- O ponto decisivo é a existência, para qualquer $[a] \subset \mathbb{Z}_m$, $[a] \neq [0]$, do elemento inverso multiplicativo $[a]^{-1}$ tal que
$$[a] \times [a]^{-1} = [a]^{-1} \times [a] = [1]$$

Demonstração

- Suponha que exista a classe $[b] = [a]^{-1}$
- Isto implicaria em $[a] \times [b] = [1]$, ou seja, m dividiria a diferença $ab - 1$
- Assim, existiria um k inteiro tal que $ab - 1 = km$
- Reescrevendo esta expressão, obtemos a equação diofantina

$$ab - km = 1,$$

  a qual só tem solução se $(a, m) = 1$
- Para que todos elementos $a = 1, 2, \ldots, m - 1$ sejam coprimos com m, m tem que ser um número primo

Inverso mutliplicativo módulo m

- Conforme vimos na demonstração anterior, o inverso multiplicativo de a módulo m só existe de a e m são coprimos
- A demonstração também nos diz como obter este inverso x: por meio da solução da equação diofantina
$$ax - km = 1$$
- Vimos anteriormente que uma solução particular x_0 esta equação pode ser obtida por meio do algorimo estendido de Euclides
- Daí, $x = [x_0]$

Proposição

Seja m um inteiro positivo tal que $m > 1$. Então $(Z_m, +, \times)$

(i) é um corpo finito, se m é primo

(ii) é um anel comutativo com unidade, se m é composto

Congruências em programação competitiva

Congruências em programação competitiva

- No contexto de programação competitiva deve-se atentar à duas propriedades fundamentais
  (i) $(a \pmod{m}) + (b \pmod{m}) \equiv a + b \pmod{m}$
  (ii) $(a \pmod{m}) \times (b \pmod{m}) \equiv ab \pmod{m}$
- Na prática, em expressões que se deseja apenas o resto do resultado por um módulo m, o módulo pode ser aplicado em todas as etapas intermediárias do cálculo
- Isto evitará o *overflow* pois, se os restos euclidianos forem utilizados como representantes das classes de equivalência, antes da extração do módulo
  (i) $a + b < 2m$, e
  (ii) $ab < m^2$

Congruências em C++

- Embora o operador % em C++ receba o nome de módulo, ele nem sempre resulta em um resto euclidiano
- As divergências surgem quando o inteiro a ou o módulo m eventualmente sejam negativos
- Para implementar o módulo compatível com o apresentado, uma alternativa é utilizar a implementação abaixo

```cpp
int mod(int a, int m)
{
    return ((a % m) + m) % m;
}
```

Adição e multiplicação modular em C++

- Este tratamento de possíveis restos negativos pode ser estendido para as operações de adição de multiplicação
- Ao se implementar métodos para a adição e para a multiplicação modulares, a leitura do código fica mais clara
- Além disso, estes métodos garantem que seus retornos serão sempre restos euclidianos
- Ao invés de utilizar o operador % duas vezes, uma condicional e uma soma são uma alternativa mais barata em termos de tempo de execução

Adição e multiplicação modular em C++

```cpp
long long add(long long a, long long b, long long m)
{
    auto r = (a + b) % m;

    return r < 0 ? r + m : r;
}

long long mul(long long a, long long b, long long m)
{
    auto r = (a * b) % m;

    return r < 0 ? r + m : r;
}
```

Teorema de Fermat, de Euler e de Wilson

Teorema de Fermat

Pequeno Teorema de Fermat

Sejam a um inteiro e p um número primo. Então

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

Fermat e inversos multiplicativos módulo p

- Se a não é um múltiplo de p, então $(a, p) = 1$, de modo que existe o inverso multiplicativo de a módulo p
- Neste caso, ao multiplicar ambos lados da congruência por a^{-1}, temos que

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

- O lado esquerdo pode ser reescrito como

$$a \times a^{p-2} \equiv 1 \pmod{p}$$

- Assim, se $(a, p) = 1$, então $a^{-1} \equiv a^{p-2} \pmod{p}$

Inversos multiplicativos módulo p em C++ com complexidade $O(\log p)$

```cpp
long long fast_exp_mod(long long a, long long n, long long m) {
    long long res = 1, base = a;

    while (n) {
        if (n & 1)
            res = mul(res, base, m);

        base = mul(base, base, m);
        n >>= 1;
    }

    return res;
}

long long inv(long long a, long long p) {
    return fast_exp_mod(a, p - 2, p);
}
```

Teorema de Euler

Teorema de Euler

Seja a e m inteiros positivos tais que $m > 1$ e $(a, m) = 1$. Então

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

Euler e inversos multiplicativos módulo m

- Nos casos em que m é primo e $(a, m) = 1$, o Pequeno Teorema de Fermat é um caso especial do Teorema de Euler
- Nos casos de módulos que não são primos, o Teorema de Euler fornece uma alternativa para o cálculo do inverso multiplicativo módulo m
- Em termos de código, ambos teoremas fornecem maneiras efetivas de computar os inversos modulares, com complexidades $O(\log m)$ e $(\log \varphi(m))$, respectivamente
- Além disso, a implementação é fácil de lembrar, enquanto que o algoritmo de Euclides estendido não é tão simples de lembrar e ainda pode gerar números negativos

Implementação do inverso multiplicativo em $O(\log \varphi(m))$

```
// É assumido que (a, m) = 1
long long inverse(long long a, long long m)
{
    return fast_exp_mod(a, phi(m) - 1, m);
}
```

Teorema de Wilson

Teorema de Wilson

Se p é um número primo, então

$$(p-1)! \equiv -1 \pmod{p}$$

Referências

1. **HALIM**, Felix; **HALIM**, Steve. *Competitive Programming 3*, 2010.
2. **HEFEZ**, Abramo. Aritmética, Coleção PROFMAT, SBM, 2016.
3. **LAAKSONEN**, Antti. *Competitive Programmer's Handbook*, 2018.
4. **SKIENA**, Steven S.; **REVILLA**, Miguel A. *Programming Challenges*, 2003.
5. Wikipédia. Fermat's little theorem, acesso em 05/04/2021.
6. Wikipédia. Euler's theorem, acesso em 05/04/2021.
7. WolframMathWorld. Wilson's Theorem, acesso em 05/04/2021.