Matemática

Divisibilidade

Prof. Edson Alves

Faculdade UnB Gama

Divisibilidade

Definição

Sejam a e b dois números inteiros. Dizemos que a **divide** b (ou que b é divisível por a) se existe um k inteiro tal que $b = ak$. Caso não exista tal inteiro, dizemos que a não divide b. Dizemos também que b é um **múltiplo** de a.

Notação: $a|b$ (lê-se "a divide b")

Relações triviais

- Qualquer número a divide a si mesmo, pois $a = a \times 1$
- Observe que 1 divide qualquer inteiro m, pois $m = 1 \times m$
- Segundo a definição de divisibilidade, zero divide zero, pois $0 = k \times 0$ para qualquer k inteiro
- De fato, qualquer inteiro a divide zero, pois $0 = a \times 0$

Unicidade de k

- Se a é diferente de zero e a divide b, então o inteiro k tal que $b = ak$ é único
- Suponha que exista um t tal que $b = ak = at$. Como a é diferente de zero, vale o cancelamento da multiplicação, de modo que $k = t$
- Observe que se $s \neq r$, ainda vale que $0 = 0 \times r = 0 \times s$
- Assim, k não fica determinado (por isso que $0/0$ é uma indeterminação)
- Para todos os demais valores $a \neq 0$, o **quociente** k de b por a é o inteiro k tal que $b = ak$

Propriedades da divisibilidade

Para quaisquer inteiros a, b e c vale que:

1. se $a|b$ e $b|c$ então $a|c$ (propriedade transitiva)
2. $a|a$ (propriedade reflexiva)
3. se $a|b$ e $b|a$, então $a = b$ ou $a = -b$
4. se $a|b$ então $|a| \leq |b|$
5. se $a|b$ e $a|c$ então $a|(bx + cy)$, para quaisquer x, y inteiros

Definição

Sejam a, b inteiros, com $b \neq 0$. Segundo a **divisão de Euclides** existem dois inteiros q, r, únicos, com $0 \leq r < |b|$, tais que $a = bq + r$. O número q é o **quociente** da divisão e r é o **resto**.

Observe que, se $r = 0$, então b divide a.

Resto da divisão em C++

O operador % (resto da divisão) em C/C++ não corresponde ao resto da divisão euclidiana em todos os casos:

```cpp
int main()
{
    int a = 11;
    int b = 7;

    cout << (a % b) << '\n';       // 4
    cout << (a % -b) << '\n';      // 4
    cout << (-a % b) << '\n';      // -4
    cout << (-a % -b) << '\n';     // -4

    return 0;
}
```

Resto da divisão em C++

- Segundo a divisão euclidiana, os quocientes e restos seriam

```
 11 = 7 x 1 + 4              // q = 1, r = 4
 11 = (-7) x (-1) + 4        // q = -1, r = 4
-11 = 7 x (-2) + 3           // q = -2, r = 3
-11 = (-7) x 2 + 3           // q = 2, r = 3
```

- Nos casos em que $a < 0$, o operador % retorna um resto negativo, o que viola a condição $0 \leq r < |b|$ da divisão de Euclides
- Para determinar o resto euclidiano nestes casos, basta somar o valor absoluto de b ao resto negativo

Maior Divisor Comum

Definição

Dados dois inteiros a e b, o **maior divisor comum** (MDC) de a e b é o inteiro não-negativo d tal que

1. d divide a e d divide b;
2. se c divide a e c divide b, então c divide d.

Notação: $d = (a, b)$

Observações sobre a definição do MDC

- A primeira condição apresentada garante que d é divisor comum de a e b
- A segunda garante que ele é o maior dentre os divisores comuns de a e b
- Pode-se observar que
  1. $d = 0$ se, e somente se, $a = b = 0$;
  2. $(a, 0) = |a|$, para todo inteiro a.
- Como $(a, b) = (-a, b) = (a, -b) = (-a, -b)$, o problema de se determinar o MDC pode ser restrito aos números não-negativos

Cálculo do MDC

- Se a e b são dois inteiros não-negativos, com $a \geq b > 0$, por Euclides existem únicos q e r tais que $a = bq + r$, com $0 \leq r < b$
- Escrevendo $r = a - bq$, é possível mostrar que $(a, b) = (b, r)$
- Lembrando que $(a, 0) = a$, o MDC pode ser computado com complexidade $O(\log a)$

```
long long gcd(long long a, long long b)
{
    return b ? gcd(b, a % b) : a;
}
```

Algoritmo de Euclides Estendido

- É possível mostrar também que o MDC entre a é b é o menor número não-negativo que pode ser escrito como uma combinação linear $ax + by$
- Esta interpretação é fundamental para a demonstração de várias propriedades associadas ao MDC
- Para se determinar tais inteiros x e y (os quais não são únicos) pode-se usar uma versão estendida do algoritmo do MDC, denominada Algoritmo de Euclides Estendido

Implementação do Algoritmo de Euclides Estendido em C++

```
long long ext_gcd(long long a, long long b, long long& x, long long& y)
{
    if (b == 0)
    {
        x = 1;
        y = 0;
        return a;
    }

    long long x1, y1;
    long long d = ext_gcd(b, a % b, x1, y1);

    x = y1;
    y = x1 - y1*(a/b);

    return d;
}
```

Equações Diofantinas Lineares

- Uma importante aplicação do MDC e do algoritmo de Euclides estendido é a solução de equações diofantinas lineares
- Para a, b, c, x, y inteiros, as equações diofantinas lineares são da forma

$$ax + by = c$$

- Tais equações tem solução se, e somente se, (a, b) divide c

Solução particular

Uma solução **particular** (x_0, y_0) de uma equação diofantina linear pode ser determinada da seguinte maneira:

1. Determine x' e y' tais que $ax' + by' = d$ (Algoritmo de Euclides estendido)
2. Faça $k = c/d$
3. Compute $x_0 = k \times x'$ e $y_0 = k \times y'$

Observe que

$$ax_0 + by_0 = a(kx') + b(ky') = k(ax' + by') = kd = c$$

Solução geral das Equações Diofantinas Lineares

- A solução particular não é única
- A solução geral de uma equação diofantina linear é dada por, para qualquer inteiro t, por

$$
\begin{aligned}
x &= x_0 + (a/d)t \\
y &= y_0 - (b/d)t
\end{aligned}
$$

- Estas expressões nos permitem determinar, por exemplo, soluções específicas, como a de menor x (ou y), menor diferença entre x e y, menor solução com x e y positivos, e assim por diante (se existirem)

Números coprimos

Definição

Dois números a e b são dito **coprimos**, ou primos entre si, se $(a, b) = 1$.

Observe que, para dois inteiros a e b quaisquer, se $d = (a, b)$, então

$$\left(\frac{a}{d}, \frac{b}{d}\right) = 1$$

Menor Múltiplo Comum

Definição

Sejam a e b dois inteiros. O **menor múltiplo comum** (MMC) de a e b é o inteiro m tal que

1. a divide m e b divide m;
2. se a divide n e b divide n, então m divide n.

Notação: $m = [a, b]$

Cálculo do MMC

- De forma similar ao MDC, a primeira propriedade torna m um múltiplo comum de a e b; a segunda o torna o menor dentre os múltiplos comuns
- Uma importante relação entre o MDC e o MMC é que $ab = (a, b)[a, b]$

```
long long lcm(long long a, long long b)
{
    return (a/gcd(a, b))*b;
}
```

- Veja que, na implementação acima, a divisão é feita antes do produto: esta ordem pode evitar *overflow* em alguns casos

Referências

1. **HALIM**, Felix; **HALIM**, Steve. *Competitive Programming 3*, 2010.
2. **HEFEZ**, Abramo. Arimética, Coleção PROFMAT, SBM, 2016.
3. **LAAKSONEN**, Antti. *Competitive Programmer's Handbook*, 2018.
4. **SKIENA**, Steven S; **REVILLA**, Miguela A. *Programming Challenges*, 2003.