Teoria dos Números - Parte 2

SCC5900 - Projeto de Algoritmos

João Batista

Máximo Divisor Comum

- Como 1 divide qualquer inteiro, então o mínimo divisor como de qualquer par de inteiros é 1.
- Mais interessante é o **máximo divisor comum**, ou **mdc**, ou seja, o maior divisor comum compartilhado por um par de inteiros.
- Diz-se que dois inteiros a e b são **relativamente primos** se o $mdc(a, b) = 1$.

Algoritmo de Euclides

- O algoritmo de Euclides para encontrar o mdc de dois inteiros é considerado o primeiro algoritmo interessante da história.
  - Outras formas seriam testar todos os divisores de *a* em *b*;
  - Ou encontrar os fatores primos de *a* e *b* e calcular o produto de todos os fatores comuns;
  - Ambas as abordagens são computacionalmente intensivas.

Algoritmo de Euclides

- O algoritmo de Euclides se baseia em duas observações:
  - Se $b | a$, então $mdc(a, b) = b$;
  - Se $a = bt + r$, então $mdc(a, b) = mdc(b, r)$.
- O algoritmo de Euclides pode ser aplicado recursivamente, substituindo $max(a, b)$ pelo resto da divisão de $max(a, b)$ por $min(a, b)$.
- O algoritmo de Euclides realiza um número logarítmico de iterações.

Algoritmo de Euclides

- Por exemplo, se *a* = 34398 e *b* = 2132:

mdc(34398, 2132) = *mdc*(34398 % 2132, 2132) = *mdc*(2132, 286)
mdc(2132, 286) = *mdc*(2132 % 286, 286) = *mdc*(286, 130)
mdc(286, 130) = *mdc*(286 % 130, 130) = *mdc*(130, 26)
mdc(130, 26) = *mdc*(130 % 26,26) = *mdc*(26, 0)

Portanto, *mdc*(34398, 2132) = 26.

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

Mínimo Múltiplo Comum

- Outra função interessante entre dois inteiro *a* e *b* é o **mínimo múltiplo comum**, **mmc**:
  - É o menor inteiro divisível por *a* e *b*;
  - Por exemplo, mmc(24, 36) = 72.
- Uma aplicação de mmc é o cálculo da periodicidade entre dois eventos periódicos distintos:
  - Qual é o próximo ano (após 2000) que a eleição presidencial (4 anos) coincidirá com o censo (10 anos)?
  - Os eventos coincidem a cada 20 anos, pois mmc(4,10) = 20.

Mínimo Múltiplo Comum

- É evidente que $mmc(a, b) \geq max(a, b)$. De forma similar, como $a \times b$ é múltiplo de ambos a e b, então $mmc(a, b) \leq ab$.
- O algoritmo de Euclides provê uma forma eficiente de calcular mmc, uma vez que $mmc(a, b) = ab/mdc(a, b)$.
  - Entretanto, é necessário ter cuidado com a possibilidade de *overflow* na multiplicação de a por b.

Aritmética Modular

- Em diversos problemas, está-se interessado conhecer o resto de divisões de inteiros:
  - Por exemplo, dado que o seu aniversário é em uma quarta-feira, quando será o seu aniversário no próximo ano?
    - Basta saber o resto da divisão de 365 (ou 366) por 7, ou seja, 365 % 7 = 1 ou 366 % 7 = 2;
  - Portanto, o aniversário pode cair um uma quinta-feira ou sexta-feira, dependendo se o ano atual é bissexto ou não.

Aritmética Modular

- **Aritmética modular** permite que diversos cálculos similares sejam feitos de forma eficiente, ou seja, sem o uso de aritmética de grandes números.
- O número dividido é chamado de **módulo** e o resto de **resíduo**.
- As operações aritméticas podem ser realizadas da seguinte maneira...

Aritmética Modular

- Adição

  $(x + y) \% n = ((x \% n) + (y \% n)) \% n$

  Quantos centavos eu tenho se receber \$123,45 da minha mãe e \$94,67 do meu pai?
  (12345 % 100) + (9467 % 100) = (45 + 67) % 100 = 12.

- Subtração

  Pode-se considerar uma adição com números negativos.
  Quantos centavos eu tenho após gastar \$52,53?
  (12 % 100) – (53 % 100) = -41 % 100 = 59 % 100.

Aritmética Modular

- Multiplicação:

  Pode-se considerar uma adição repetida.

  $xy \% n = (x \% n)(y \% n) \% n$.

  Quantos centavos você terá se receber \$17.28 por hora com 2143 horas trabalhadas?

  $(1728 \times 2143) \% 100 = (1728 \% 100) \times (2143 \% 100) = (28*43)\%100 = 1204\%100 = 4$

- Exponenciação:

  $x^y \% n = (x \% n)^y \% n$

Aritmética Modular

- Aritmética modular possui diversas aplicações interessantes:
  - Encontrar os últimos dígitos.
  - Algoritmo de criptografia RSA.
  - Cálculos de Calendário.