Algoritmo de Karatsuba e Teorema Mestre

Prof. André Vignatti

Problema: multiplicar dois inteiros x e y de n bits

- x_{L}: os $\frac{n}{2}$ bits mais à esquerda de x em binário
- x_{R}: os $\frac{n}{2}$ bits mais à direita de x em binário
- y_{L} e y_{R} definidos de maneira análoga

Assim,

$$x = 2^{n/2}x_{\text{L}} + x_{\text{R}}, \quad (1)$$
$$y = 2^{n/2}y_{\text{L}} + y_{\text{R}} \quad (2)$$

Usando (1) e (2),

$$xy = (2^{n/2}x_{\text{L}} + x_{\text{R}})(2^{n/2}y_{\text{L}} + y_{\text{R}}) = 2^n x_{\text{L}}y_{\text{L}} + 2^{n/2}(x_{\text{L}}y_{\text{R}} + x_{\text{R}}y_{\text{L}}) + x_{\text{R}}y_{\text{R}}$$

Assim, para calcular xy, dependemos de quatro produtos: $x_{\text{L}}y_{\text{L}}$, $x_{\text{L}}y_{\text{R}}$, $x_{\text{R}}y_{\text{L}}$, $x_{\text{R}}y_{\text{R}}$
Depois, combinamos isso somando três termos:

(a) $2^n x_{\text{L}}y_{\text{L}}$

(b) $2^{n/2}(x_{\text{L}}y_{\text{R}} + x_{\text{R}}y_{\text{L}})$

(c) $x_{\text{R}}y_{\text{R}}$

Portanto, computar xy se resume ao custo de:

- Quatro multiplicações $x_{\text{L}}y_{\text{L}}$, $x_{\text{L}}y_{\text{L}}$, $x_{\text{L}}y_{\text{L}}$, $x_{\text{L}}y_{\text{L}}$ (todas com números de $\frac{n}{2}$ bits)
- Computar (a) usando n deslocamentos à esquerda
- Computar (b) fazendo uma soma usando $n/2$ deslocamentos à esquerda
- Somar (a), (b), (c)

Assim, temos o seguinte algoritmo de divisão e conquista:

Algoritmo 1: $Mult(x, y)$

if $n = 1$ **then**
  $z = x \wedge y$
else
  $z_{\text{LL}} = \text{Mult}(x_{\text{L}}, y_{\text{L}})$
  $z_{\text{LR}} = \text{Mult}(x_{\text{L}}, y_{\text{R}})$
  $z_{\text{RL}} = \text{Mult}(x_{\text{R}}, y_{\text{L}})$
  $z_{\text{RR}} = \text{Mult}(x_{\text{R}}, y_{\text{R}})$
  $z = \text{Reconstroi}(z_{\text{LL}},\ z_{\text{LR}},\ z_{\text{RL}},\ z_{\text{RR}})$
return z

Algoritmo 2: $Reconstroi(z_{\text{LL}},\ z_{\text{LR}},\ z_{\text{RL}},\ z_{\text{RR}})$

$a = \text{ShiftLeft}(z_{LL}, n)$
$b = z_{LR} + z_{RL}$
$b = \text{ShifLeft}(b,\ \frac{n}{2})$
$z = a + b + z_{RR}$
return z

A função *ShiftLeft(z, k)* retorna $z(\underbrace{0 \ldots 0}_{k \text{ bits}})$ (ou seja, multiplica z por 2^k)

ShiftLeft(z, k) leva tempo $\Theta(k)$. A soma $x + y$ leva tempo $\Theta(n)$.

O custo total do algoritmo é $T(n) = 4T(\frac{n}{2}) + \Theta(n)$.

Algoritmo de Karatsuba

Reescrevendo xy:

$$\begin{aligned} xy &= 2^n x_{\text{L}} y_{\text{L}} + 2^{n/2}(x_{\text{L}} y_{\text{R}} + x_{\text{R}} y_{\text{L}}) + x_{\text{R}} y_{\text{R}} \\ &= 2^n x_{\text{L}} y_{\text{L}} + 2^{n/2}[(x_{\text{L}} + x_{\text{R}})(y_{\text{L}} + y_{\text{R}}) - x_{\text{L}} y_{\text{L}} - x_{\text{R}} y_{\text{R}})] + x_{\text{R}} y_{\text{R}} \end{aligned}$$

A vantagem é computar apenas três produtos:

- $x_{\text{L}} y_{\text{L}}$
- $(x_{\text{L}} + x_{\text{R}})(y_{\text{L}} + y_{\text{R}})$
- $x_{\text{R}} y_{\text{R}}$

Algoritmo 3: $Karatsuba(x, y)$

if $n = 1$ **then**
  $z = x \wedge y$
else
  $a' = x_{\text{L}} + x_{\text{R}}$
  $a'' = y_{\text{L}} + y_{\text{R}}$
  $z_{\text{LL}} = \text{Karatsuba}(x_{\text{L}}, y_{\text{L}})$
  $a \quad = \text{Karatsuba}(a', a'')$
  $z_{\text{RR}} = \text{Karatsuba}(x_{\text{R}}, y_{\text{R}})$
  $z = \text{ReconstroiKaratsuba}(z_{\text{LL}}, a, z_{\text{RR}})$
return z

Algoritmo 4: ReconstroiKaratsuba(z_{LL}, a, z_{RR}))

$b = a - z_{\text{LL}} - z_{\text{RR}}$
$z' = \text{ShifLeft}(z_{\text{LL}}, n)$
$z'' = \text{ShifLeft}(b, \frac{n}{2})$
return $z' + z'' + z_{\text{RR}}$

O custo total do algoritmo de Karatsuba é $T(n) = 3T(\frac{n}{2}) + \mathcal{O}(n)$.

Teorema Mestre

Teorema. *Sejam* $a \geq 1$, $b > 1$, $T, f : \mathbb{N} \to \mathbb{R}$ *tais que*

$$T(n) = aT(n/b) + f(n),$$

então

$$T(n) = \begin{cases} \Theta(n^{\log_b a}), & se\ f(n) = \mathcal{O}(n^{\log_b a - \epsilon}), \epsilon > 0, \\ \Theta(n^{\log_b a} \log n), & se\ f(n) = \Theta(n^{\log_b a}), \\ \Theta(f(n)), & se\ f(n) = \Omega(n^{\log_b a + \epsilon}), \epsilon > 0\ e \\ & af(n/b) < cf(n)\ assintoticamente\ c < 1. \end{cases}$$

Aplicando na primeira recorrência, temos $T(n) = \Theta(n^2)$. Aplicando na recorrência do algoritmo de Karatsuba, $T(n) = \Theta(n^{\lg 3})$. Podemos dizer também que:

- $T(n) = \mathcal{O}(n^{1.59})$
- $T(n) = \Omega(n^{1.58})$

pois $1.58 < \lg 3 < 1.59$