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,

$$
\begin{aligned}
x &= 2^{n/2} x_{\mathrm{L}} + x_{\mathrm{R}}, \\
y &= 2^{n/2} y_{\mathrm{L}} + y_{\mathrm{R}}
\end{aligned}
$$

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

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

Assim, para calcular xy, dependemos de quatro produtos:

$$\underbrace{x_{\mathrm{L}}y_{\mathrm{L}},\ x_{\mathrm{L}}y_{\mathrm{R}},\ x_{\mathrm{R}}y_{\mathrm{L}},\ x_{\mathrm{R}}y_{\mathrm{R}}}_{n/2 \text{ bits}}$$

Depois, combinamos isso somando três termos:

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

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

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

Algoritmo 1: *Mult*(x, y)

if $n = 1$ **then**

$\quad z = x \wedge y$

else

$\quad z_{\mathrm{LL}} = \mathrm{Mult}(x_{\mathrm{L}}, y_{\mathrm{L}})$

$\quad z_{\mathrm{LR}} = \mathrm{Mult}(x_{\mathrm{L}}, y_{\mathrm{R}})$

$\quad z_{\mathrm{RL}} = \mathrm{Mult}(x_{\mathrm{R}}, y_{\mathrm{L}})$

$\quad z_{\mathrm{RR}} = \mathrm{Mult}(x_{\mathrm{R}}, y_{\mathrm{R}})$

$\quad z = \mathrm{Reconstroi}(z_{\mathrm{LL}},\ z_{\mathrm{LR}},\ z_{\mathrm{RL}},\ z_{\mathrm{RR}})$

return z

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

$a = \mathrm{ShiftLeft}(z_{LL}, n)$

$b = z_{LR} + z_{RL}$

$b = \mathrm{ShifLeft}(b,\ \frac{n}{2})$

$z = a + b + z_{RR}$

return z

$\textit{ShiftLeft}(z,k)$ retorna $z(\underbrace{0\ldots0}_{k \text{ bits}})$ (ou seja, multiplica z por 2^k)

$\textit{ShiftLeft}(z,k)$ leva tempo $\Theta(k)$

A soma $x+y$ leva tempo $\Theta(n)$

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

Reescrevendo xy:

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

$$= 2^n x_{\mathrm{L}} y_{\mathrm{L}} + 2^{n/2}[(x_{\mathrm{L}} + x_{\mathrm{R}})(y_{\mathrm{L}} + y_{\mathrm{R}}) - x_{\mathrm{L}} y_{\mathrm{L}} - x_{\mathrm{R}} y_{\mathrm{R}})] + x_{\mathrm{R}} y_{\mathrm{R}}$$

A vantagem é computar apenas três produtos:

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

Algoritmo 3: *Karatsuba*(x, y)

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

  $z = \mathrm{ReconstroiKaratsuba}(z_{\mathrm{LL}}, a, z_{\mathrm{RR}})$
return z

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

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

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

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

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

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 $1.58 < \lg 3 < 1.59$