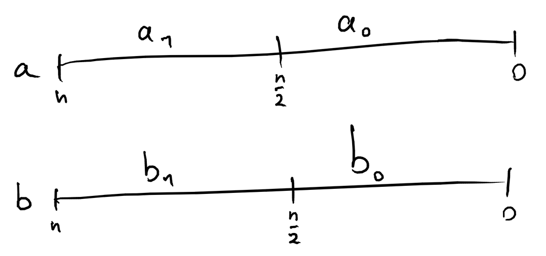
[1h skip]

Mnożenie liczb

Naiwnie: $O(n^2)$

Algorytm Karatsuby

Pomysł 1



$(n=2^k)$

$$a = a_1 \cdot 2^{\frac{n}{2}} + a_0$$

$$b = b_1 \cdot 2^{\frac{n}{2}} + b_0$$

$$a \cdot b = a_1 \cdot b_1 \cdot 2^n + (a_0 b_1 + a_1 b_0) 2^{\frac{n}{2}} + a_0 b_0$$

$$T(n) = 4 \cdot T(\tfrac{n}{2}) + \Theta(n)$$

$$T(n) = \Theta(n^2) \quad :c \qquad n^{\log_2 4}$$

Pomysł 2

$$a \cdot b = \underbrace{a_1 \cdot b_1}_{c_2} \cdot 2^n + \underbrace{(a_0 b_1 + a_1 b_0)}_{c_1} 2^{\frac{n}{2}} + \underbrace{a_0 b_0}_{c_0}$$

$w_0 = a_0 b_0$

$w_1 = a_1 b_1$

$w_2 = (a_0 + a_1)(b_1 + b_0) = a_0 b_1 + a_0 b_0 + a_1 b_1 + a_1 b_0$

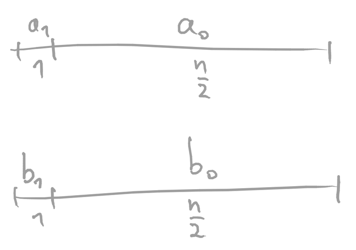
$\frac{n}{2}+1$ bitów?

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \bar{w} = \bar{c}$$

$c_0 = w_0$

$c_1 = w_2 - w_0 - w_1$

$c_2 = w_1$



$$a \cdot b = \underbrace{a_1 b_1 2^n + (a_1 b_0 + b_1 a_0) 2^{\frac{n}{2}}}_{O(n)} + \underbrace{a_0 b_0}_{T(\frac{n}{2})}$$

$$T(n) = 3\,T(\tfrac{n}{2}) + \Theta(n)$$

$$T(n) = \Theta(n^{\log_2 3})$$

$$T(n) = k\,T(\tfrac{n}{k}) + \Theta(n) \rightarrow T(n) = n \log n$$

Czy możemy lepiej?

Dzielimy na k części

przykład: $k=3$

$(n=3^k)$

$$a = a_0 + a_1 2^{\frac{n}{3}} + a_2 2^{\frac{2n}{3}}$$

$$b = b_0 + b_1 2^{\frac{n}{3}} + b_2 2^{\frac{2n}{3}}$$

$$a \cdot b = \underbrace{a_0 b_0}_{c_0} + 2^{\frac{n}{3}} \underbrace{(a_0 b_1 + b_0 a_1)}_{c_1} + 2^{\frac{2n}{3}} \underbrace{(a_0 b_2 + b_0 a_2 + a_1 b_1)}_{c_2} +$$

$$+ 2^n \underbrace{(a_1 b_2 + b_1 a_2)}_{c_3} + 2^{\frac{4n}{3}} \cdot \underbrace{a_2 b_2}_{c_4}$$

$w_0 = a_0 b_0$

$w_1 = (a_0 + a_1 + a_2)(b_0 + b_1 + b_2)$

$w_2 =$

$w_3 =$

$w_4 = a_2 b_2$

Sumy:

- a_0
- a_1
- a_2
- b_0
- b_1
- b_2
- $a_0 + a_1 + a_2$
- $b_0 + b_1 + b_2$

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ & & & & \\ & & & & \\ & & & & \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ & -1 & 1 & -1 & 1 \\ & & 1 & -1 & -1 \\ 0 & & & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & \cdots \\ 1 & 4 & 16 & 64 & \cdots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- dynamiczne dopasowywanie podziału