

# ALGORITMER (DATASTRUKTURER) OCH KOMPLEXITET

## FÖRELÄSNING

- ENVARIABELPOLYNOM
  - OLIKA LAGERINGSSÄTT
  - ADDITION, MULTIPLIKATION
  - EVALUERING, INTERPOLATION
- DFT - DISKRET FOURIERTRANSFORM
- FFT - SNABB FOURIERTRANSFORM

# OPERATIONER PÅ ENVARIABELPOLYNOM

POLYNOMET  $A(x) = \sum_{j=0}^{n-1} a_j x^j = a_0 + a_1 x + \dots + a_{n-1} x^{n-1}$

KAN LAGRAS I KOEFFICIENTFORM SOM

$$\langle a_0, a_1, \dots, a_{n-1} \rangle$$

EVALUERING:

MED HORNERS REGEL:

$$A(x) = a_0 + x(a_1 + x(a_2 + \dots + x(a_{n-2} + x \cdot a_{n-1}) \dots))$$

Tid:  $\Theta(n)$

ADDITION:

$$\sum_{j=0}^{n-1} a_j x^j = \sum_{j=0}^{n-1} b_j x^j + \sum_{j=0}^{n-1} c_j x^j$$

BERÄKNAS GENOM  $a_j = b_j + c_j$  FÖR  $0 \leq j \leq n-1$

Tid:  $\Theta(n)$

MULTIPLIKATION:

$$\sum_{j=0}^{2n-2} a_j x^j = \left( \sum_{j=0}^{n-1} b_j x^j \right) \cdot \left( \sum_{j=0}^{n-1} c_j x^j \right)$$

BERÄKNAS GENOM  $a_j = \sum_{k=0}^j b_k \cdot c_{j-k}$

Tid:  $\Theta(n^2)$



# ALTERNATIV LAGRINGSMETOD

POLYNOMET  $A(x) = \sum_{j=0}^{n-1} a_j x^j$  KAN OCKSÅ

LAGRAS I PUNKT-VÄRDE-FORM SOM

$\{(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})\}$  DÄR  $y_j = A(x_j)$

ENTYDIGT, TY BARA ETT  $(n-1)$ -GRADSPOLYNOM  
GÅR GENOM  $n$  GIVNA PUNKTER [NUMMEKURSEN]

INTERPOLATION (ÖVERGÅNG TILL KOEFFICIENTFORM):

MED LAGRANGES INTERPOLATIONSFORMEL

$$A(x) = \sum_{k=0}^{n-1} y_k \frac{\prod_{j \neq k} (x - x_j)}{\prod_{j \neq k} (x_k - x_j)}$$

TID:  $\Theta(n^2)$

ADDITION:

$\{(x_0, y_0 + y_0'), (x_1, y_1 + y_1'), \dots, (x_{n-1}, y_{n-1} + y_{n-1}')\}$

TID:  $\Theta(n)$

MULTIPLIKATION:

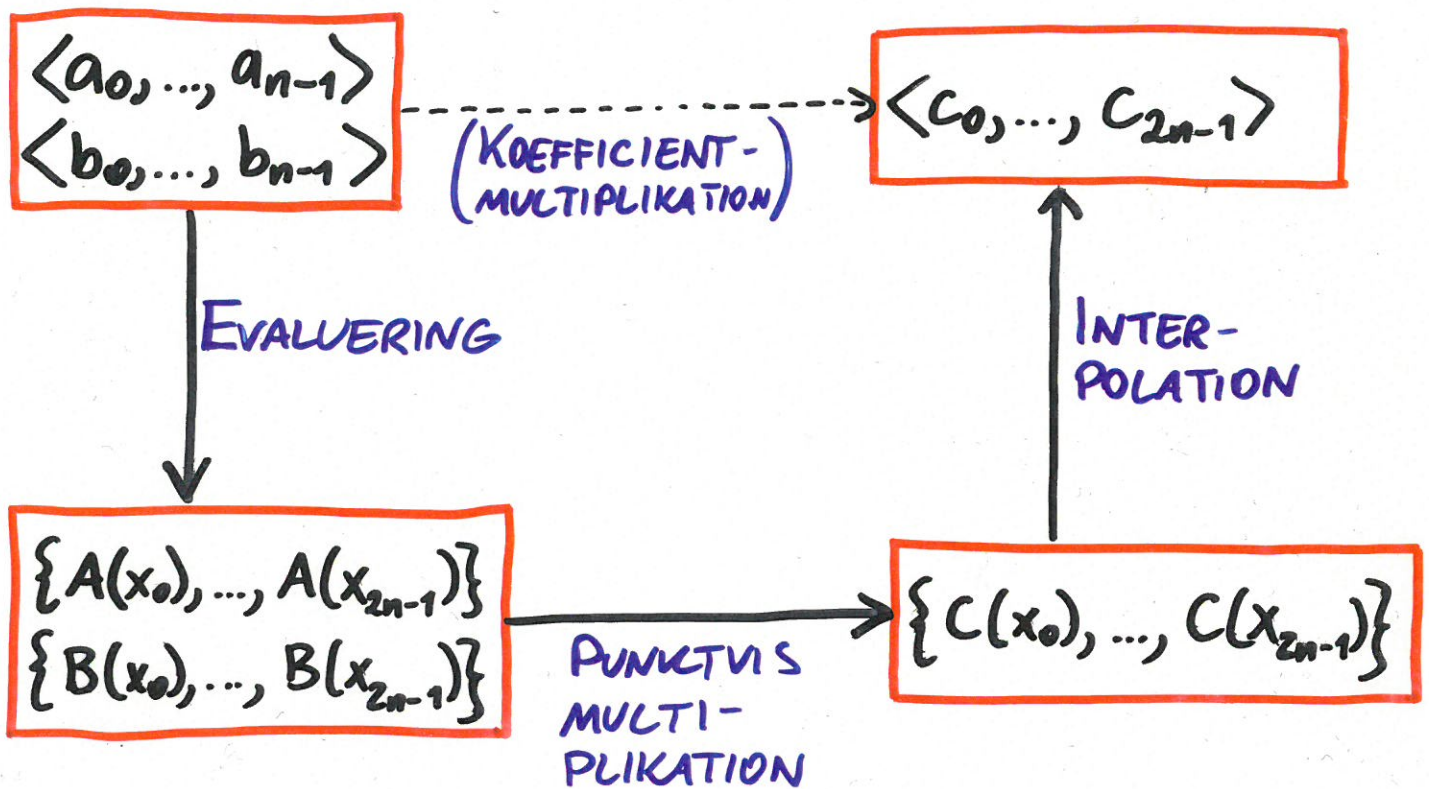
$\{(x_0, y_0 \cdot y_0'), (x_1, y_1 \cdot y_1'), \dots, (x_{2n-2}, y_{2n-2} \cdot y_{2n-2}')\}$

TID:  $\Theta(n)$

PROBLEM:  $2n-1$  PUNKTER KRÄVS.

# SNABB POLYNOMMULTIPLIKATION I KOEFFICIENTFORM

OM MAN KUNDE EVALUERA OCH INTERPOLERA  
SNABBT SKULLE MAN KUNNA MULTIPLICERA  
SNABBT MED HJÄLP AV PUNKTVIS MULTIPLIKATION.



OM EVALUERING OCH INTERPOLATION GICK ATT GÖRA  
I TID  $O(n^2)$  SKULLE DETTA VARA SNABBARE  
ÄN VANLIG POLYNOMMULTIPLIKATION.

VÄLJ EVALUERINGS-PUNKTERNA  $x_0, \dots, x_{2n-1}$  LISTIGT!



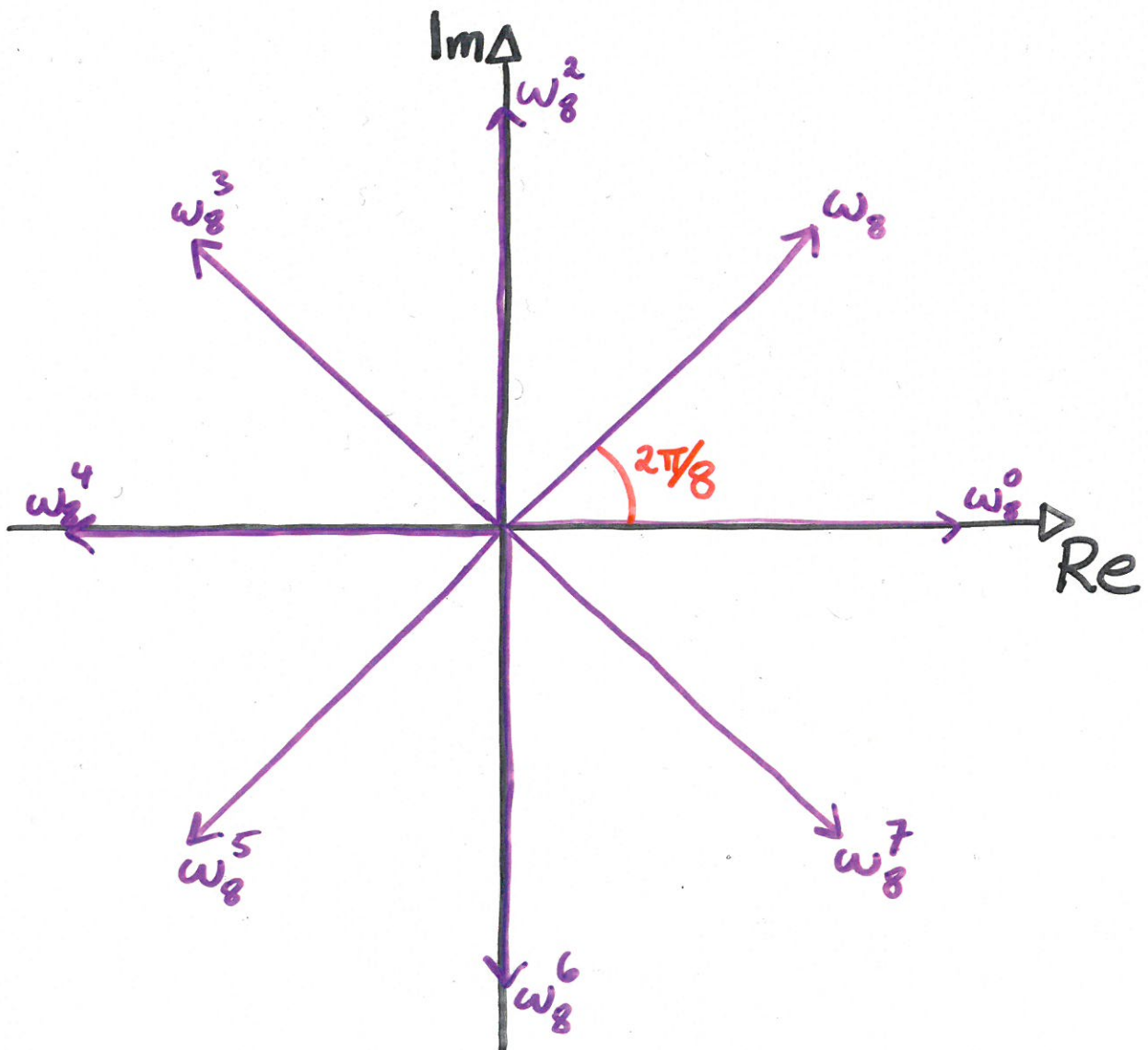
# KOMPLEXA ENHETSROTTER

VÄL ATT EVALUERA POLYNOMEN I  
KOMPLEXA ENHETSROTTER, DVS

$$\omega \text{ SÅ ATT } \omega^n = 1$$

PRINCIPALROTEN  $\omega_n = e^{\frac{2\pi i}{n}}$

ANVÄND  $\omega_n^0, \omega_n^1, \omega_n^2, \dots, \omega_n^{n-1}$



# DFT

## DISKRET FOURIERTRANSFORM

$$A(x) = \sum_{j=0}^{n-1} a_j x^j$$

POLYNOM VARS KOEFFICIENTER ÄR  $\langle a_0, a_1, \dots, a_{n-1} \rangle$

$$\text{DFT}_n(\langle a_0, a_1, \dots, a_{n-1} \rangle) = \langle y_0, y_1, \dots, y_{n-1} \rangle$$

$$\text{DÄR } y_k = A(\omega_n^k) = \sum_{j=0}^{n-1} a_j \omega_n^{jk} = \sum_{j=0}^{n-1} a_j e^{2\pi i j k / n}$$

$\text{DFT}_n$  TRANSFORMERAR EN  $n$ -VEKTOR TILL EN ANNAN  $n$ -VEKTOR.

JÄMFÖR DEN VANLIGA FOURIERTRANSFORMEN SOM TRANSFORMERAR EN FUNKTION  $f(x)$  TILL EN ANNAN FUNKTION  $\hat{f}(t)$ :

$$\hat{f}(t) = \int_{-\infty}^{\infty} f(x) e^{-itx} dx$$

# FFT - SNABB BERÄKNING AV DFT

$$\begin{aligned} \text{DELA UPP } A(x) &= \sum_{j=0}^{n-1} a_j x^j = \\ &= a_0 + a_2 x^2 + a_4 x^4 + \dots + a_{n-2} x^{n-2} + \\ &+ x(a_1 + a_3 x^2 + a_5 x^4 + \dots + a_{n-1} x^{n-2}) = \\ &= A^{[0]}(x^2) + x \cdot A^{[1]}(x^2) \end{aligned}$$

$$\begin{aligned} \text{DÄR } A^{[0]}(x) &= a_0 + a_2 x + a_4 x^2 + \dots + a_{n-2} x^{\frac{n}{2}-1} \\ A^{[1]}(x) &= a_1 + a_3 x + a_5 x^2 + \dots + a_{n-1} x^{\frac{n}{2}-1} \end{aligned}$$

$$\text{DFT}_n(\langle a_0, a_1, \dots, a_{n-1} \rangle) = \langle y_0, y_1, \dots, y_{n-1} \rangle \quad \text{DÄR}$$

$$y_k = \text{DFT}_{\frac{n}{2}}(\langle a_0, a_2, \dots, a_{n-2} \rangle)_k + \omega_n^k \text{DFT}_{\frac{n}{2}}(\langle a_1, a_3, \dots, a_{n-1} \rangle)_k$$

$$\text{DFT}_1(\langle a \rangle) = a \cdot \omega_1^0 = a$$

DEKOMPOSITION!



# FFT-ALGORITHM

ANTA ATT  $n$  ÄR EN TVÅPOTENS.

$DFT_n(\langle a_0, a_1, \dots, a_{n-1} \rangle) =$

IF  $n=1$  THEN RETURN  $\langle a_0 \rangle$

$\omega_n \leftarrow e^{2\pi i/n}$

$\omega \leftarrow 1$

$y^{[0]} \leftarrow DFT_{\frac{n}{2}}(\langle a_0, a_2, \dots, a_{n-2} \rangle)$

$y^{[1]} \leftarrow DFT_{\frac{n}{2}}(\langle a_1, a_3, \dots, a_{n-1} \rangle)$

FOR  $k \leftarrow 0$  TO  $\frac{n}{2} - 1$  DO

$y_k \leftarrow y_k^{[0]} + \omega \cdot y_k^{[1]}$

$y_{k+\frac{n}{2}} \leftarrow y_k^{[0]} - \omega \cdot y_k^{[1]}$

$\omega \leftarrow \omega \cdot \omega_n$

RETURN  $\langle y_0, y_1, \dots, y_{n-1} \rangle$

ANALYS:

$$T(n) = \begin{cases} \Theta(1) & \text{om } n=1 \\ 2T(\frac{n}{2}) + \Theta(n) & \text{om } n>1 \end{cases}$$

$$\Rightarrow T(n) = \Theta(n \log n)$$



# INVERS TILL DFT

$$\bar{y} = \text{DFT}_n(\bar{a}) \Leftrightarrow \bar{y} = V_n \bar{a} \quad \text{DVS}$$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} \omega_n^0 & \omega_n^0 & \omega_n^0 & \dots & \omega_n^0 \\ \omega_n^0 & \omega_n^1 & \omega_n^2 & \dots & \omega_n^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega_n^0 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \dots & \omega_n^{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

$V_n = \text{VANDERMONDEMATRISEN}$

$$V_n[k, j] = \omega_n^{kj}$$

VI SÖKER  $\bar{a} = \text{DFT}_n^{-1}(\bar{y}) \Leftrightarrow \bar{a} = V_n^{-1} \bar{y}$

SATS:  $V_n^{-1}[j, k] = \frac{1}{n} \omega_n^{-kj}$

$$\text{DFT}_n^{-1}(\langle y_0, y_1, \dots, y_{n-1} \rangle) = \langle a_0, a_1, \dots, a_{n-1} \rangle$$

$$\text{DÄR } a_j = \frac{1}{n} \sum_{k=0}^{n-1} y_k \omega_n^{-kj} = \frac{1}{n} \sum_{k=0}^{n-1} y_k e^{-2\pi i j k / n}$$

JÄMFÖR INVERSEN TILL DEN VANLIGA FOURIERTRANSFORMEN:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(t) e^{itx} dt$$

# POLYNOMMULTIPLIKATION - FFT

$$\sum_{j=0}^{2n-1} c_j x^j = \left( \sum_{j=0}^{n-1} a_j x^j \right) \cdot \left( \sum_{j=0}^{n-1} b_j x^j \right)$$

EVALUERING

$$\langle y_0, \dots, y_{2n-1} \rangle = \text{DFT}_{2n}(\langle a_0, \dots, a_{n-1}, 0, \dots, 0 \rangle)$$

$$\langle y'_0, \dots, y'_{2n-1} \rangle = \text{DFT}_{2n}(\langle b_0, \dots, b_{n-1}, 0, \dots, 0 \rangle)$$

$$\langle c_0, \dots, c_{2n-1} \rangle = \text{DFT}_{2n}^{-1}(\langle \underbrace{y_0 \cdot y'_0, \dots, y_{2n-1} \cdot y'_{2n-1}} \rangle)$$

INTERPOLATION

PUNKTVIS  
MULTIPLIKATION

Tid:  $\Theta(n \log n)$