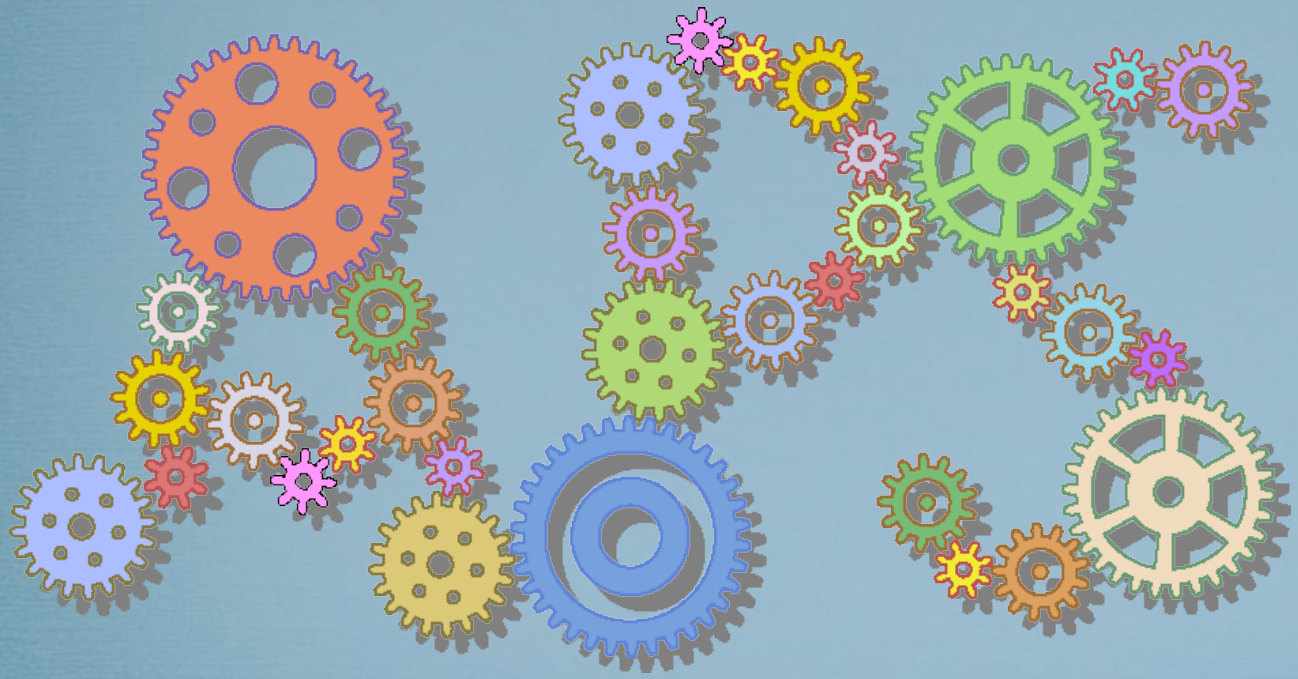
Algoritmi in podatkovne strukture



Predstavitev polinomov

- Koeficientna predstavitev.

$$a(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$$

Predstavitev polinomov

- Koeficientna predstavitev.

$$a(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$$

- Vrednostna predstavitev.
  - Nabor točk: $x = (x_0, x_1, \ldots, x_{n-1})$
  - Vrednosti v točkah: $y = (y_0, y_1, \ldots, y_{n-1})$

$$y_i = a(x_i)$$

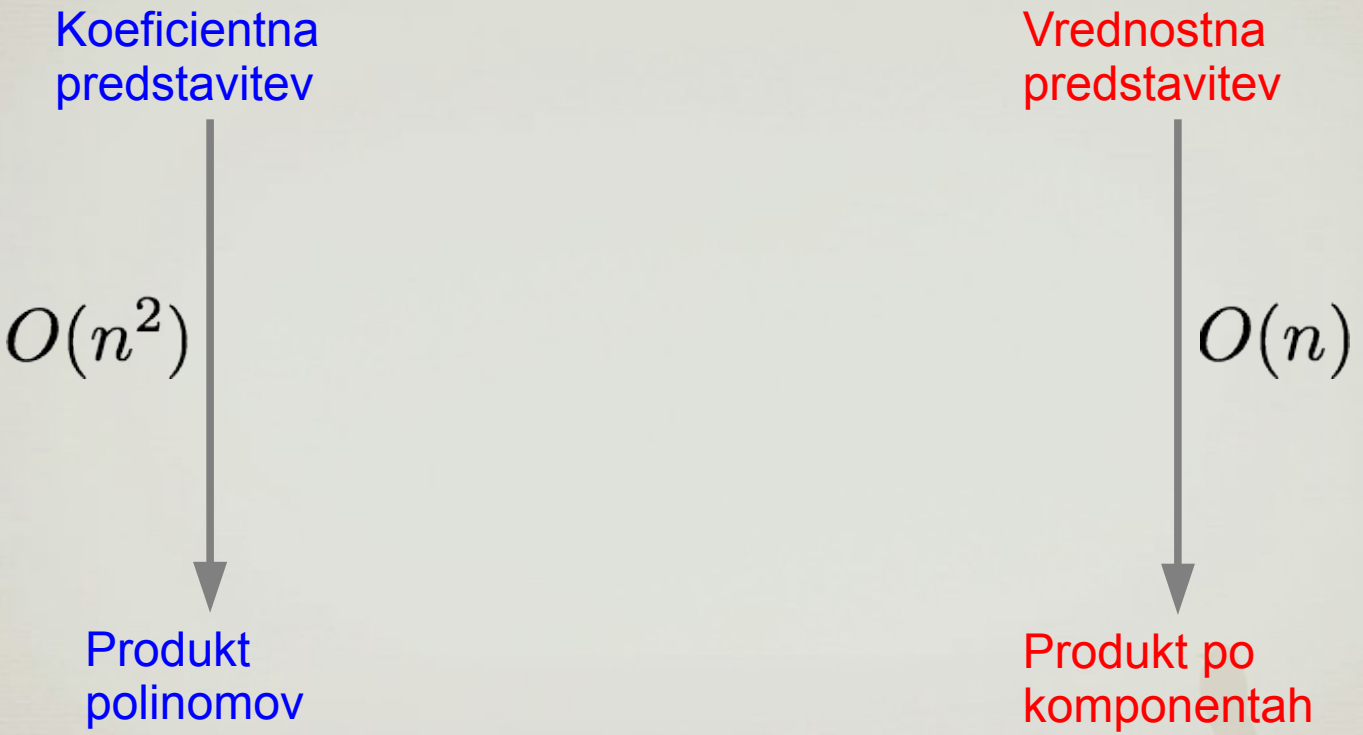
Prehod med predstavitvami

- Iz koeficientne v vrednostno.
  - Izračun vrednosti polinoma v točki – $O(n)$.
    - Uporaba Hornerjevega algoritma.
  - Izračun v n točkah – $O(n^2)$.
- Iz vrednostne v koeficientno.
  - Interpolacija s polinomom skozi n točk.

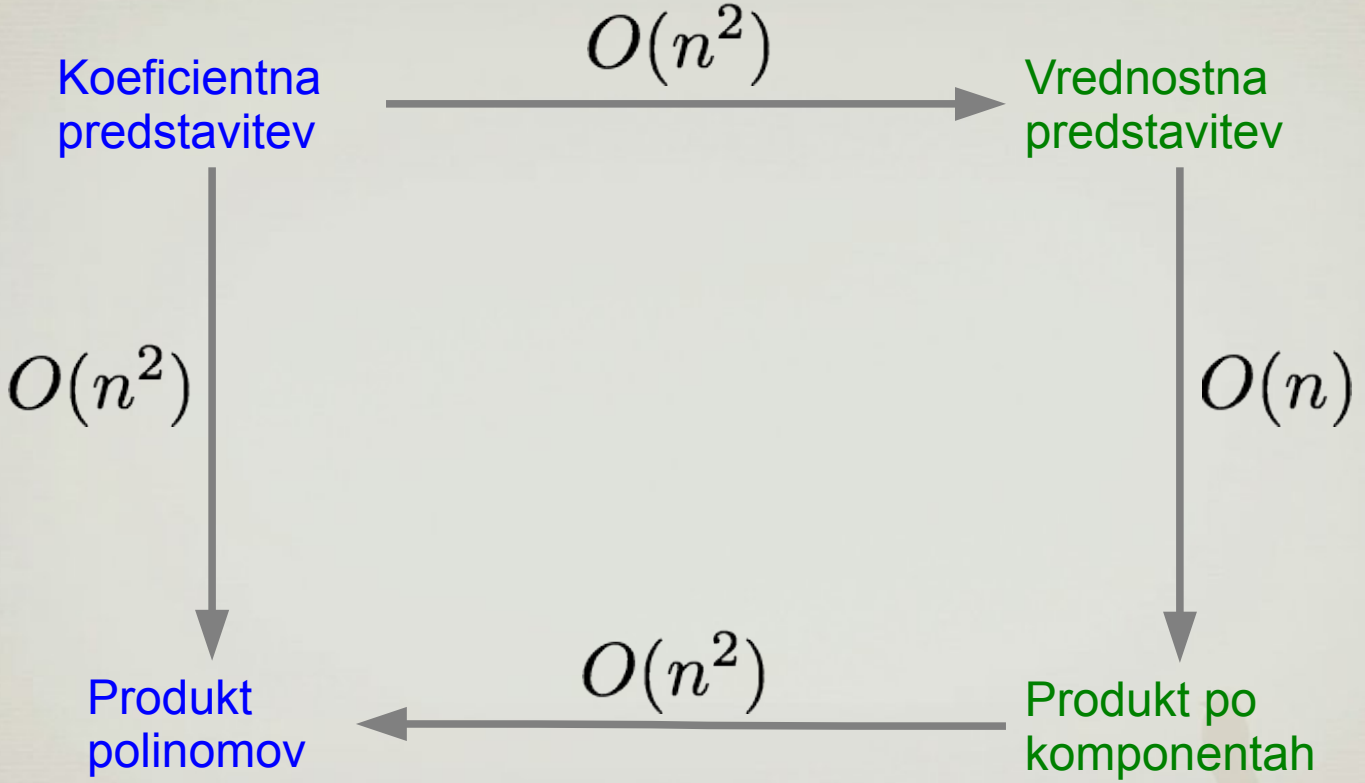
Množenje polinomov

- Množenje polinomov oz. konvolucija vektorjev.
  - Koeficientna predstavitev – $O(n^2)$.
  - Vrednostna predstavitev – $O(n)$.
- Kako to izkoristiti?
  - Polinome v koeficientni predstavitvi želimo hitreje množiti preko množenja v vrednostni predstavitvi.

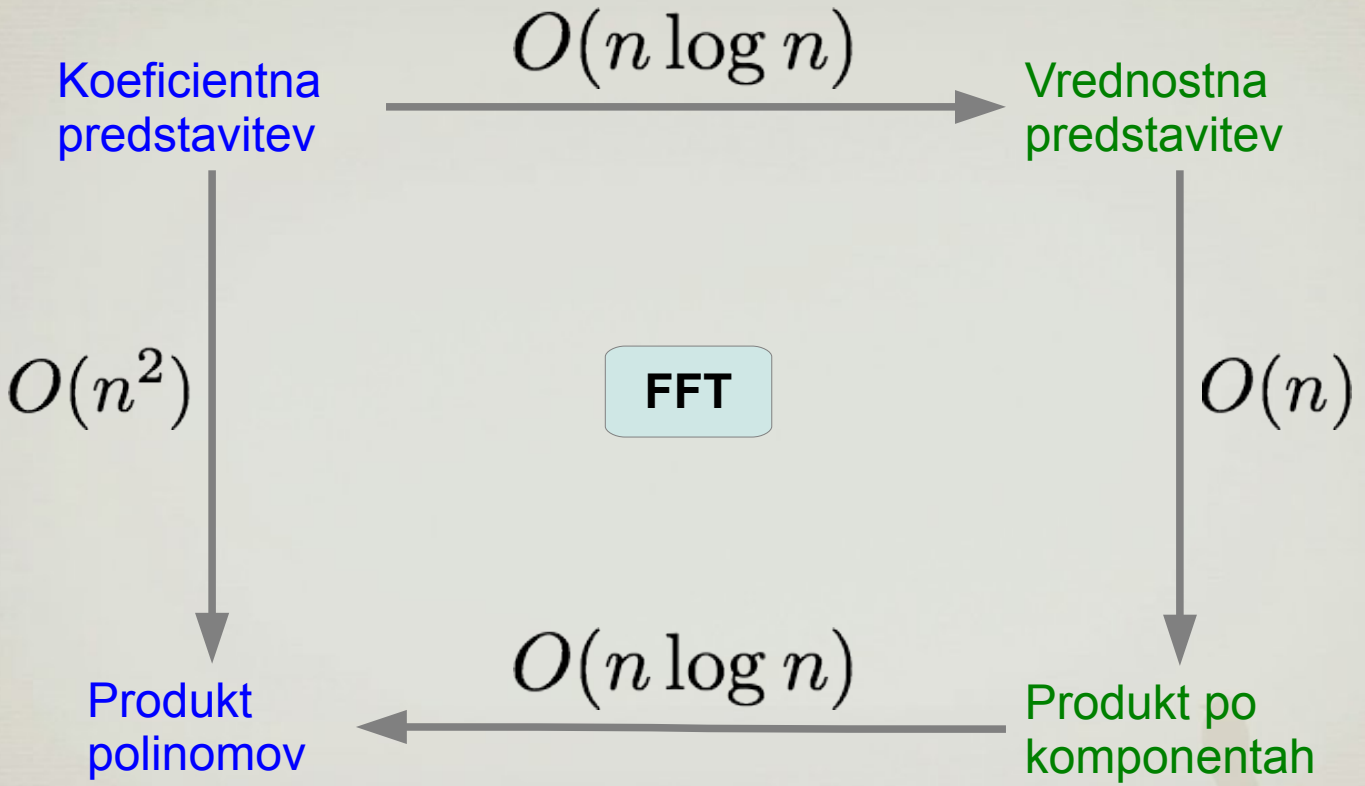
Množenje polinomov



Množenje polinomov



Množenje polinomov



Primitivni koren enote

- Komutativni kolobar (npr. kompleksna števila).
- *n*-ti primitivni koren enote:

$$\omega^n = 1$$ koren enote

$$\omega^i \neq 1 \qquad 1 \leq i < n$$ primitivni koren

Primer: kompleksna števila

- *n*-ti primitivni koren:

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

- Eulerjeva formula:

$$\omega = \cos\frac{2\pi}{n} + i\sin\frac{2\pi}{n}$$

Primer: kompleksna števila

- Dokaz.

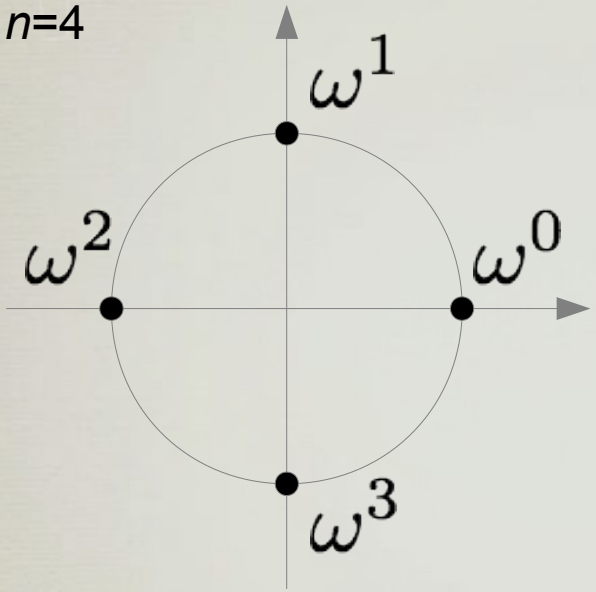
$$\omega^n = \left(e^{i\frac{2\pi}{n}}\right)^n = \cos 2\pi + i \sin 2\pi = 1$$

$$\omega^j = \left(e^{i\frac{2\pi}{n}}\right)^j = \cos 2\pi\frac{j}{n} + i \sin 2\pi\frac{j}{n} \neq 1$$

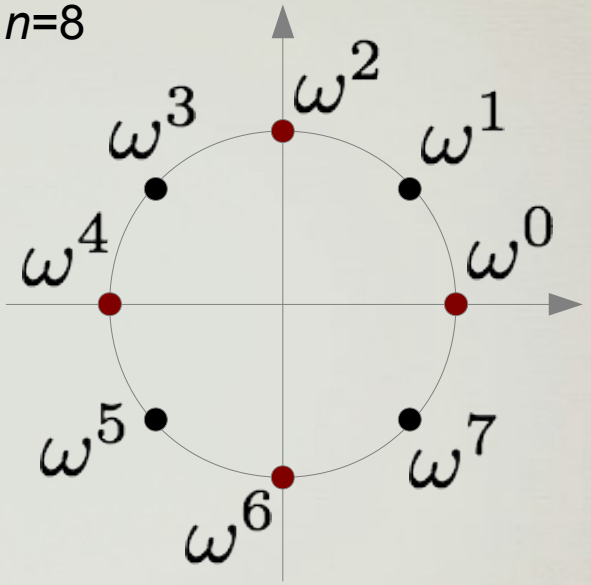
$$1 \leq j < n$$

Primer: kompleksna števila

n=4



n=8



Primer: obseg Z_p

$$(\mathbb{Z}_5, +, \times, 0, 1)$$

$$1^4 = 2^4 = 3^4 = 4^4 = 1$$

$1^1 = 1$ ✗

$2^1 = 2$ $\quad 2^2 = 4$ $\quad 2^3 = 3$ ✓

$3^1 = 3$ $\quad 3^2 = 4$ $\quad 3^3 = 2$ ✓

$4^1 = 4$ $\quad 4^2 = 1$ ✗

Diskretna Fourierjeva transformacija

- DFT je transformacija polinoma (oz. vektorja):
  - iz koeficientne predstavitve
    v vrednostno predstavitev v točkah $\omega^0, \omega^1, \ldots, \omega^{n-1}$

Vrednostna predstavitev

Koeficientna predstavitev

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = F \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

DFT

- Matrika *F*.

$$F = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega & \dots & \omega^{n-1} \\ 1 & \omega^2 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \dots & \omega^{(n-1)^2} \end{bmatrix}$$

$$F = [w^{kj}]$$

Inverzni DFT

- Inverzni DFT je skoraj enak DFT.

$$F^{-1} = \frac{1}{n}\left[\omega^{-kj}\right]$$

Množenje z *1/n.*

ω^{-1} je tudi *n*-ti PKE.

Časovna zahtevnost

- Računanje DFT z matriko.
  - Časovna zahtevnost *O(n^2)*.
  - Množenja polinomov na ta način ne pohitrimo (celo poslabšamo ga).
- Zaradi *lepih* lastnosti PKE,
  je mogoče izvajanje DFT bistveno pohitriti.
  - Algoritem FFT – $O(n \log n)$
  - O tem več naslednjič …