Algoritmi in podatkovne strukture

Hitra Fourierjeva transformacija (FFT)

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 – $O(n \log n)$

Rekurzivni FFT

- Deli in vladaj algoritem.
  - Deli
    - Delitev polinoma na sodi in lihi polinom.
    - 1x n → 2x n/2
  - Rekurzija
    - Izračun FFT na sodem in lihem polinomu.
  - Vladaj
    - Združitev FFTjev sodega in lihega polinoma.
    - 2x n/2 → 1x n

Deli

Naloga: polinom velikosti n

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

Delitev naloge

Podnalogi: sodi in lihi polinom velikosti n/2

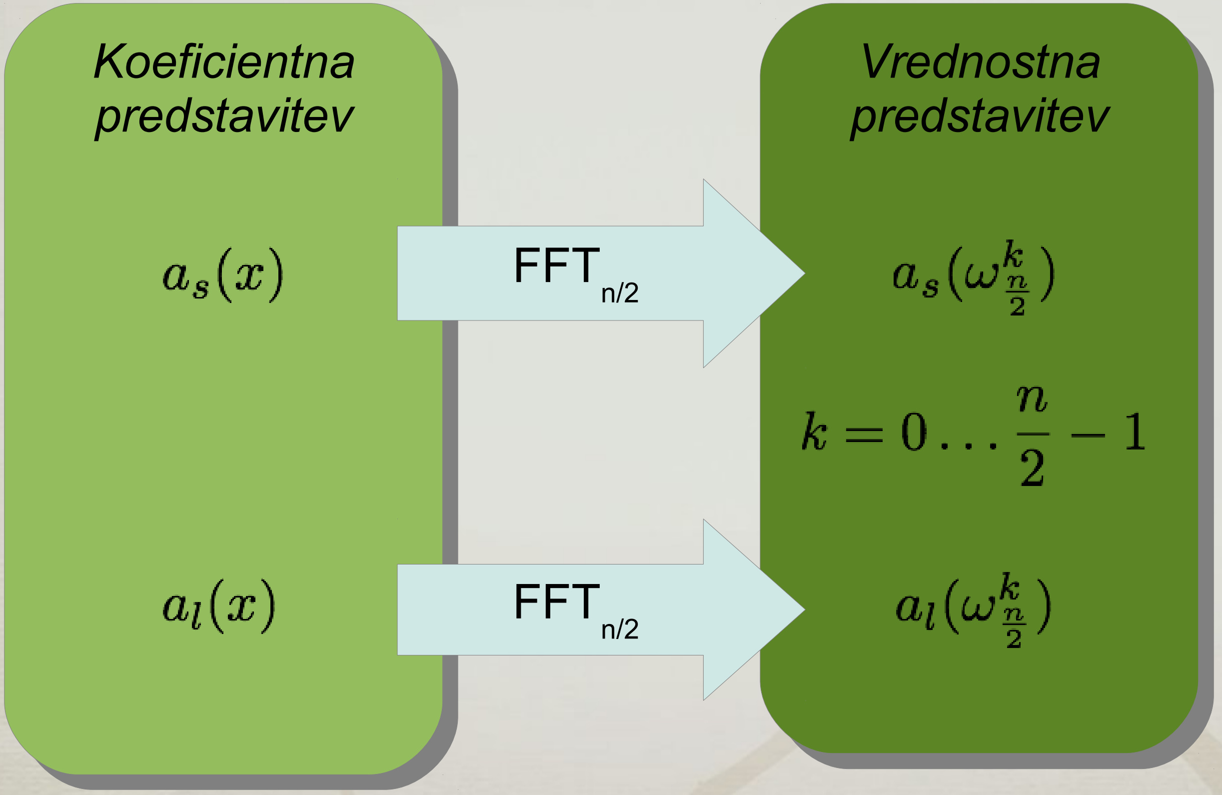
$$a(x) = a_s(x^2) + x \cdot a_l(x^2)$$

$$a_s(x) = \sum_{j=0}^{\frac{n}{2}-1} a_{2j} x^j$$

$$a_l(x) = \sum_{j=0}^{\frac{n}{2}-1} a_{2j+1} x^j$$

Rekurzija

- FFT sodega in lihega polinoma.



Vladaj

- Iz FFTjev sodega in lihega polinoma sestavimo FFT originalnega polinoma.

$$a(x) = a_s(x^2) + x \cdot a_l(x^2)$$

$$\omega_n^k \qquad k = 0 \ldots n-1$$

$$a(\omega_n^k) = a_s(\omega_n^{2k}) + \omega_n^k \cdot a_l(\omega_n^{2k})$$

Vladaj – 1. del

$$a(\omega_n^k) = a_s(w_n^{2k}) + \omega_n^k \cdot a_l(w_n^{2k}) =$$
$$= a_s(w_{\frac{n}{2}}^k) + \omega_n^k \cdot a_l(w_{\frac{n}{2}}^k)$$

$$k = 0 \ldots \frac{n}{2} - 1$$

Cancelation lemma

$$\omega_{dn}^{dk} = \omega_n^k$$

Vladaj – 2. del

$$a(\omega_n^{k+\frac{n}{2}}) = a_s((\omega_n^{k+\frac{n}{2}})^2) + \omega_n^{k+\frac{n}{2}} \cdot a_l((\omega_n^{k+\frac{n}{2}})^2) =$$
$$= a_s(\omega_n^{2k}) + \omega_n^{k+\frac{n}{2}} \cdot a_l(\omega_n^{2k}) =$$
$$= a_s(\omega_{\frac{n}{2}}^{k}) + \omega_n^{k+\frac{n}{2}} \cdot a_l(\omega_{\frac{n}{2}}^{k}) =$$
$$= a_s(\omega_{\frac{n}{2}}^{k}) - \omega_n^{k} \cdot a_l(\omega_{\frac{n}{2}}^{k})$$

$$k = 0 \ldots \frac{n}{2} - 1$$

$$(\omega^{k+\frac{n}{2}})^2 = \omega^{2k+n} =$$
$$= \omega^{2k}$$

$$\omega_{dn}^{dk} = \omega_n^k$$

$$\omega^{k+\frac{n}{2}} = -\omega^k$$

Vladaj – povzetek

$$a(\omega_n^k) = a_s(w_{\frac{n}{2}}^k) + \omega_n^k \cdot a_l(w_{\frac{n}{2}}^k)$$

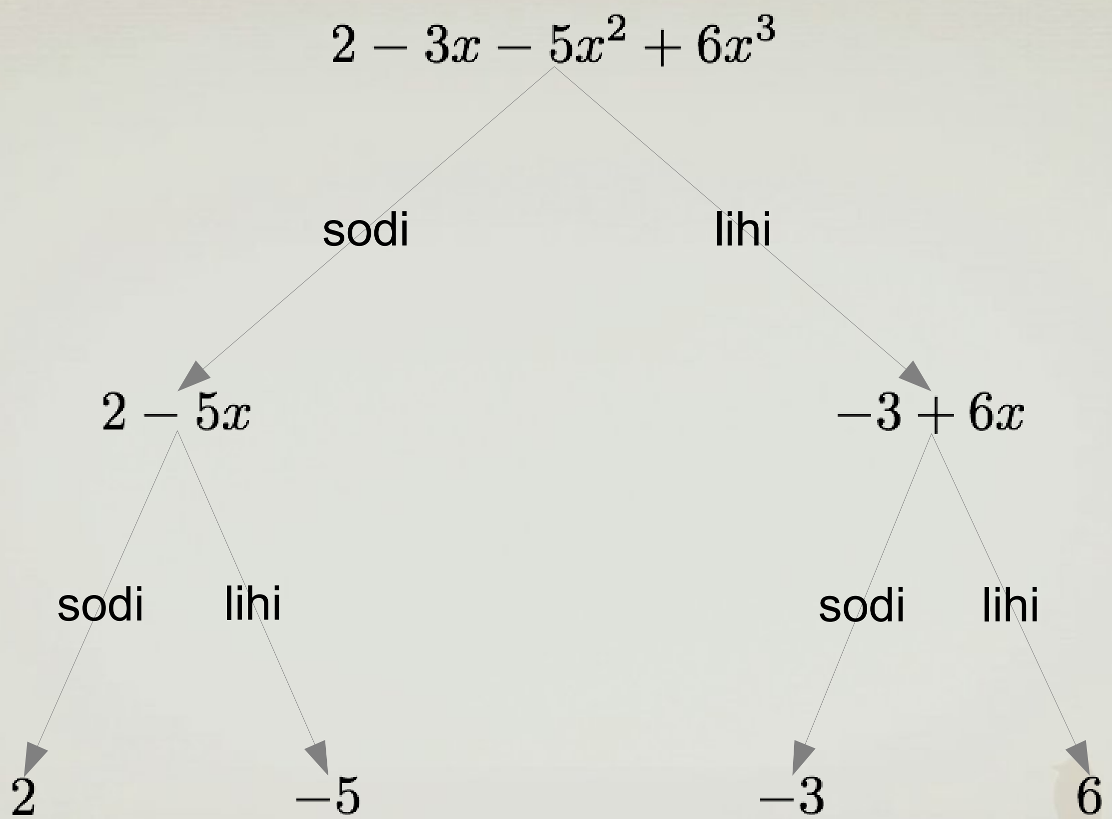
$$a(\omega_n^{k+\frac{n}{2}}) = a_s(\omega_{\frac{n}{2}}^k) - \omega_n^k \cdot a_l(\omega_{\frac{n}{2}}^k)$$

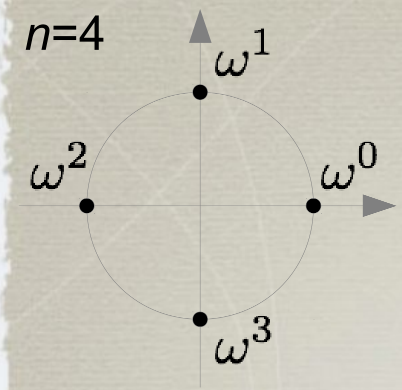
$$k = 0 \ldots \frac{n}{2} - 1$$

Algoritem

```
fun recursiveFFT(a) is
    n = a.length
    if n == 1 then return a

    ys = recursiveFFT([a_0, a_2, …, a_n-2])
    yl = recursiveFFT([a_1, a_3, …, a_n-1])

    w = e^(2 PI i / n)
    wk = 1
    y = [0, 0, …, 0]
    for k = 0 to n/2-1 do
        y[k]       = ys[k] + wk * yl[k]
        y[k+n/2] = ys[k] - wk * yl[k]
        wk = wk * w
    end

    return y
end
```

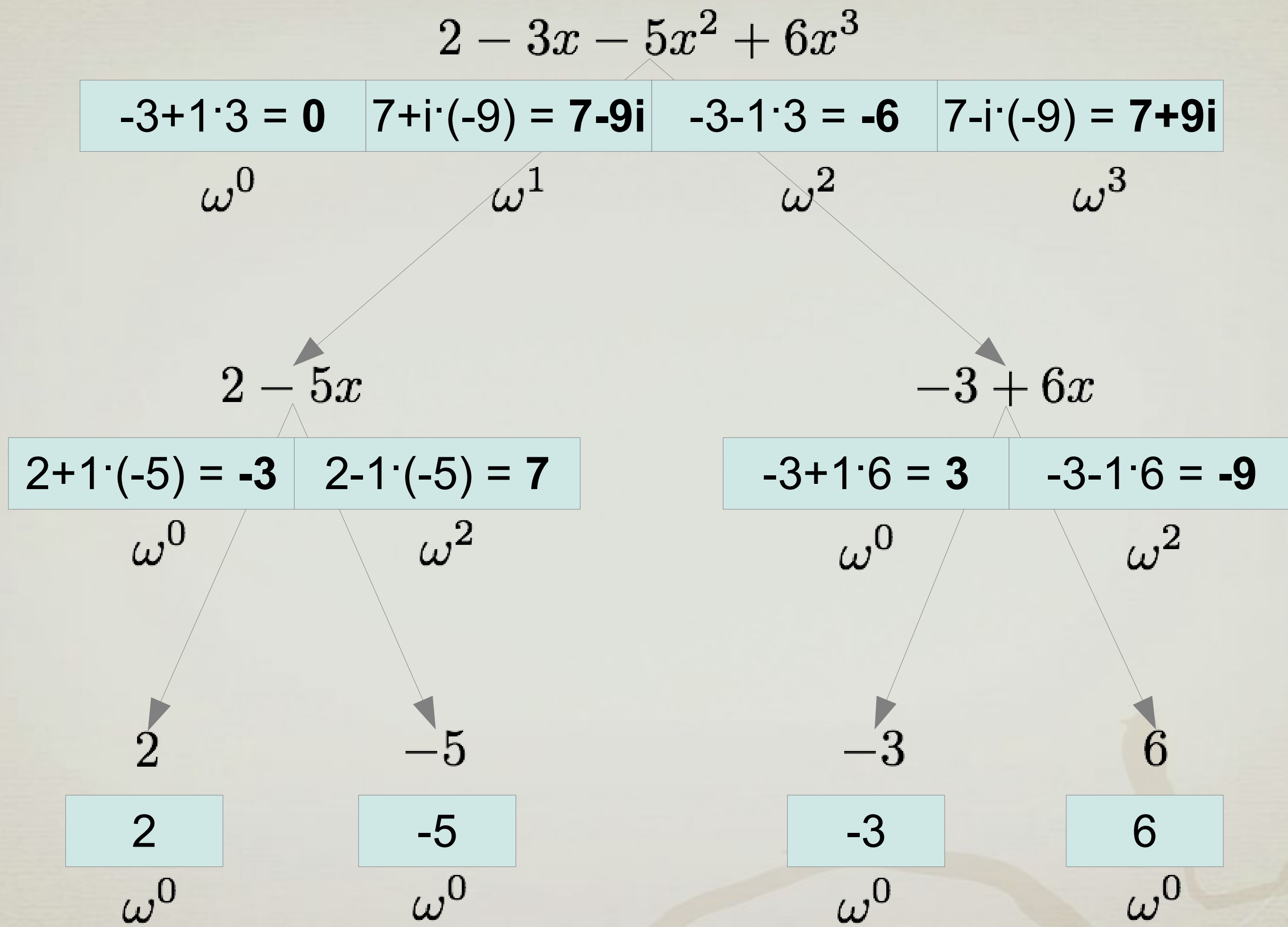
Primer 1



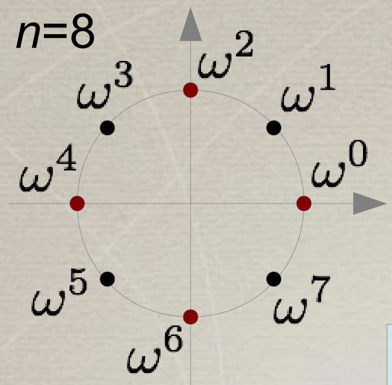


n=4
ω^1
ω^2
ω^0
ω^3

Primer 1



$2 - 3x - 5x^2 + 6x^3$
-3+1·3 = **0**
7+i·(-9) = **7-9i**
-3-1·3 = **-6**
7-i·(-9) = **7+9i**
ω^0
ω^1
ω^2
ω^3
$2 - 5x$
$-3 + 6x$
2+1·(-5) = **-3**
2-1·(-5) = **7**
-3+1·6 = **3**
-3-1·6 = **-9**
ω^0
ω^2
ω^0
ω^2
2
-5
-3
6
2
-5
-3
6
ω^0
ω^0
ω^0
ω^0



Primer 2

$$3 + 5x + 4x^2 + 0x^3 - 2x^4 + 5x^5 + x^6 + 0x^7$$

ω^0	ω^1	ω^2	ω^3				
16	5+3i	-4+10i	5-3i	-4	5+3i	-4-10i	5-3i

$$3 + 4x - 2x^2 + x^3$$

ω^0	ω^2		
6	5+3i	-4	5-3i

$$5 + 0x + 5x^2 + 0x^3$$

ω^0	ω^2		
10	0	10	0

$$3 - 2x$$

ω^0	
1	5

$$4 + x$$

ω^0	
5	3

$$5 + 5x$$

ω^0	
10	0

$$0 + 0x$$

ω^0	
0	0

$3 \quad -2 \quad 4 \quad 1 \quad 5 \quad 5 \quad 0 \quad 0$

3	-2	4	1	5	5	0	0