HITRA FOURIEROVA TRANSFORMACIJA

FFT(koef):

sodi = FFT(sodi koeficienti)
lihi = FFT(lihi koeficienti)

prckanje = $[e^{\frac{2\pi i}{N}k}$ for $k \in [0, \dots, \frac{N}{2}-1]]$

join(sodi + prckanje · lihi,
sodi − prckanje · lihi)

---

1) Naj bo $\omega = e^{\frac{2\pi i}{n}}$ koren enote in n sodo število. Dokažite:

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

b) $\omega^a = \omega^b$ če $a \equiv b \mod n$

c) $\frac{1}{\omega^k} = \overline{\omega}^k$

a) $\omega^{k+\frac{n}{2}} = \omega^k \omega^{\frac{n}{2}} = \omega^k e^{\frac{2\pi i}{n} \cdot \frac{n}{2}} = \omega^k e^{\pi i} = -\omega^k$

b) $b \equiv a \mod n \iff b = n \cdot k + a$

$\omega^b = (\omega^n)^k \omega^a = (\omega^{2\pi i})^k \omega^a = 1 \cdot \omega^a = \omega^a$

c) $\omega^{-k} = (e^{-\frac{2\pi i}{n}})^k = (\overline{e^{\frac{2\pi i}{n}}})^k = \overline{\omega}^k$

DISKRETNA FOURIEROVA TRANSFORMACIJA

$$p^*[j] = \sum_{k=0}^{n-1} p[k]\, \omega_n^{jk}$$

polinom z vrednostmi v točkah v korenih enote

koeficienti polinoma

$$p[j] = \frac{1}{n} \sum_{k=0}^{n-1} p^*[k]\, \omega_n^{-kj}$$

---

2) Za diskretno fourierovo transformacijo dokažite:

a) Linearnost

b) Inverznost

a) p, q polinoma

$$(\alpha p + \beta q)^*[j] = \sum_{k=0}^{n-1} (\alpha p + \beta q)[k]\, \omega_n^{jk}$$

$$= \alpha \sum p[k]\, \omega^{jk} + \beta \sum q[k]\, \omega_n^{jk}$$

$$= \alpha p^*[j] + \beta q^*[j]$$

b) $$p[j] = \frac{1}{n} \sum_{k=0}^{n-1} \Big(\sum_{\ell=0}^{n-1} p[\ell]\, \omega_n^{\ell j} \Big) \omega_n^{-jk}$$

$$= \frac{1}{n} \sum_{\ell=0}^{n-1} p[\ell] \sum_{k=0}^{n-1} \omega_n^{k(\ell - j)}$$

$$= \frac{1}{n} \sum_{\ell=0}^{n-1} p[\ell] \cdot \begin{cases} \dfrac{\overbrace{1-\omega_n^{(\ell-j)n}}^{0}}{1-\omega_n^{(\ell-j)}} = 0 & ; \quad \ell \neq j \\ n & ; \quad \ell = j \end{cases}$$

$= \frac{1}{n} \cdot n\, p[j]$

$= p[j]$

Končnost + Linearnost

$\Rightarrow$ (Lev inverz $\Rightarrow$ Inverz)

✓

---

3) Izpelji rekurzivno zvezo za FFT iz definicije.

$$p^*[j] = \sum p[k]\, \omega_n^{jk}$$

$$= \sum_{l=0}^{\frac{n}{2}-1} \left(p[2l]\, \omega_n^{2lj} + p[2l+1]\, \omega_n^{2lj} \right)$$

$$= \sum_{l=0}^{\frac{n}{2}-1} \left(p[2l]\, \omega_{\frac{n}{2}}^{lj} + \omega_n^{j}\, p[2l+1]\, \omega_{\frac{n}{2}}^{lj} \right)$$

$$= p_s^*[j] + \omega_n^j\, p_l^*[j]$$

Množenje polinomov s FFT je $O(n \log n)$.

---

3) Imamo 2 vektorjev dolžine n in računamo vse skalarne produkte $\langle a, b[i:] + b[:i] \rangle$.

$$p = \sum_{i=0}^{n-1} a_i x^i$$

$$q = \sum_{j=0}^{n-1} b_j x^j$$

$(a_0+a_1x+\dots+a_{n-1}x^{n-1})\cdot(b_0+b_1x+\dots+b_{n-1}x^{n-1})$

$1: a_0 \cdot b_0$
$x: a_0b_1+a_1b_0$
$x^2: a_0b_2+a_1b_1+a_2b_0$
$\vdots$
$x^k: \sum_{i=0}^{k} a_i b_{k-i}$

$\langle a,b\rangle = \sum a_i b_i$

$\langle a, b_j\rangle = \sum a_i b_{i+j}$

zamik

Torej:

$g(x) = \sum_{i=0}^{n-1} b_{n-1-i}\, x^i$

$p(x) = \sum_{i=0}^{n-1} a_i x^i$

1	a_0b_3			
x	a_0b_2	a_1b_3		
x^2	a_0b_1	a_1b_2	a_2b_3	
x^3	a_0b_0	a_1b_1	a_2b_2	a_3b_3
x^4		a_1b_0	a_2b_1	a_3b_2
x^5			a_2b_0	a_3b_1
x^6				a_3b_0

$= \langle a,b\rangle$

Algoritem:

i) Obrni b

ii) Zmnoži p, g $\quad (O(n\log n))$

iii) $p \cdot q$ = rezultat

iv) Za dan i dobimo $\langle a, b[i:] + b[:i] \rangle$

Če je $i = 0$, je to $p \cdot q[n-1]$

Če je $i = 1$, je to $p \cdot q[n-2] + p \cdot q[2n-2]$

Splošno: $p \cdot q[n-i-1] + p \cdot q[2n-1-i]$

---

4) Izpelji imperativno implementacijo FFT.

```
def FFT(koef):

    n = len(koef)

    if n == 1:
        return koef

    assert n % 2 == 0

    sodi_koef = koef[::2]
    lihi_koef = koef[1::2]

    sodi_koef = FFT(sodi_koef)
    lihi_koef = FFT(lihi_koef)
```

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

$w = 1$

```
rezultat = [0] * n

for k in range(n/2):
    rezultat[k] = sodi_koef[k] + w * lihi_koef[k]
    rezultat[n/2 + k] = sodi_koef[k] - w * lihi_koef[k]
    w = w * w_n

return rezultat
```

19.3.

ZVEZNA FOURIEROVA TRANSFORMACIJA

$$\hat{f}(\hat{x}) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \hat{x}}\, dx$$

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\hat{x})\, e^{2\pi i x \hat{x}}\, d\hat{x}$$

---

1) Naj bosta A, B končni množici, $\max(A), \max(B) \leq M$.

Izpelji $O(M \log M)$ algoritem, ki prešteje, na koliko možnih načinov lahko dobimo vse možne izide, torej na koliko načinov lahko dobimo vsako od možnih vsot $a+b$, $a \in A$, $b \in B$.

Primer: $A = \{0,1,2\}$
$B = \{0,2\}$

$[1,1,2,1,1]$

$A \mapsto \vec{a} \in \{0,1\}^{M+1}$

$$a_i = \begin{cases} 1 \; ; \; i \in A \\ 0 \; ; \; i \notin A \end{cases}$$

$IFFT(FFT_{2M}(\vec{a}) \cdot FFT(\vec{b}))$

---

2) Naj bo $B \subseteq \mathbb{Z}$, $|B| = m$.

V $O(n^2)$ ugotovi, ali B vsebuje tri različne elemente, ki se seštejejo v 0.

```
for x ∈ B:
    for y ∈ B:
        preveri -x-y ∈ B\{x,y}
```

---

3) Naj bo $B \subseteq \mathbb{Z}$, $|B| = m$ in naj za nek $n \in \mathbb{N}$ velja $|x| \leq n$ za vse $x \in B$.

V $O(n \log n)$ ugotovi, ali B vsebuje tri različne elemente, ki se seštejejo v 0.

$$a_i = \begin{cases} 1; & i-n \\ 0; & \text{sicer} \end{cases}$$

$$\vec{b} = [a_0 0\ a_1 0\ a_2 0 \cdots a_n 0]$$

$$\vec{c} = [a_0 0 0\ a_1 0 0\ a_2 0 0 \cdots a_n 0 0]$$

$$IFFT(FFT(\vec{a}) \cdot FFT(\vec{a}) \cdot FFT(\vec{a}))$$

$$-3\, IFFT(FFT(\vec{b}) \cdot FTT(\vec{a}))$$

$$+2\, \cancel{IFTT}(\cancel{FFT}(\vec{c}))$$