Diskretna Fourierova Transformacija – DFT

Želimo izračunati vrijednost polinoma A, reda n (stupnja $n-1$)

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

u svim n-tim korijenima iz jedinice, tj. u točkama

$$\omega_n^0, \omega_n^1, \ldots, \omega_n^{n-1} \qquad \left(\begin{array}{l} z_k = \omega_n^k \\ k=0,\ldots,n-1 \end{array} \right)$$

gdje je

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

Kasnije ćemo uzeti da je n potencija od 2, $n = 2^m$, ali trenutno sve vrijedi za bilo koji n.

Tj. treba izračunati vektor $y = (y_0, \ldots, y_{n-1})^T$ vrijednosti

$$y_k = A(\omega_n^k) = \sum_{j=0}^{n-1} a_j \omega_n^{k \cdot j}, \quad k = 0, \ldots, n-1.$$

Vektor vrijednosti $y = (y_0, \ldots, y_{n-1})^T$ zovemo diskretna Fourierova transformacija vektora koeficijenata $a = (a_0, \ldots, a_{n-1})^T$. Zapis je

$$y = \mathrm{DFT}_n(a)$$

$n \to$ označava duljinu (dimenziju) uzorka – vektora.

Ovu operaciju možemo pisati i u matričnom obliku

$$y = V_n \cdot a$$

gdje V_n matrica reda n, s elementima

$$(V_n)_{kj} = \omega_n^{kj}, \quad j, k = 0, \ldots, n-1$$

$\hookrightarrow$ nije baš uobičajeno da su indeksi iz $\{0, \ldots, n-1\}$, već iz $\{1, \ldots, n\}$, ali je ovdje puno zgodnije.

Uočimo odmah da je V_n specijalna Vandermondeova matrica ($x_j = \omega_n^j$).

Vidimo odmah da je V_n simetrična (kompleksna) matrica
$$V_n = V_n^T$$
(ali nije Hermitska).

Njezin inverz se lako računa iz svojstava n-tih korijena iz jedinice.

Za $n \in \mathbb{N}$ i $k \in \mathbb{N}_0$, ako n ne dijeli k, onda je
$$\sum_{j=0}^{n-1} (\omega_n^k)^j = 0 \qquad = \left|\frac{k}{n}\right|, \text{ ako } n \text{ dijeli } k \; (k \in \mathbb{Z}).$$
Koristeći ovo, lako se dokazuje da za inverz V_n^{-1} vrijedi
$$V_n^{-1} = \frac{1}{n} \cdot V_n^{*}$$
(tj. $U_n := \frac{1}{\sqrt{n}} V_n$
$U_n^{-1} = \sqrt{n} \cdot V_n^{-1} = \frac{\sqrt{n}}{n} \cdot V_n^{*}$
$= \frac{1}{\sqrt{n}} V_n^{*} = (U_n)^{*}$
$\Rightarrow U_n$ ortogonalna ili, preciznije, unitarna!)

Posebno,
$$\overline{\omega_n} = \omega_n^{-1}.$$
Po elementima je:
$$(V_n^{-1})_{kj} = \frac{1}{n} \omega_n^{-kj}.$$
Inverzna diskretna Fourierova transformacija je onda
$$a = DFT_n^{-1}(y)$$
ili
$$a = V_n^{-1} y$$
tj. ako DFT ide s ω_n onda DFT^{-1} ide s ω_n^{-1} i još, na kraju, pomnoži sve $\frac{1}{n}$

odnosno, po elementima
$$a_j = \frac{1}{n} \cdot \sum_{k=0}^{n-1} y_k \, \omega_n^{-k \cdot j}, \quad j = 0, \dots, n-1.$$

Nap: - kako se drukčije definiraju DFT i DFT^{-1}
- periodičnost
- grupa n-tih korijena iz jed. (mult.) $\approx (\mathbb{Z}_n, +_n)$ $\qquad \omega_n^k = \omega_n^{k \bmod n}, \; \forall k \in \mathbb{Z}$.
- halving lemma
[n paran, $\omega_n^2 = \omega_{n/2}$] $\qquad \omega_{dn}^{dk} = \omega_n^k, \; n \geq 0, \; k \geq 0, \; d > 0$
- koef. op's = elementary!

$\omega_n^{n/2} = -1$, n par

$(\omega_n^{k+\frac{n}{2}})^2 = (\omega_n^k)^2$, n par

Brzi rekurzivni algoritam FFT za DFT_n, ako je $n = 2^m$, $m \in \mathbb{N}_0$, tj. n je potencija od 2, dobivamo rastavom polinoma A

$$A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots + a_{n-1} x^{n-1}$$

na parne i neparne koeficijente:

$$A(x) = (a_0 + a_2 x^2 + \dots + a_{n-2} x^{n-2}) + x \cdot (a_1 + a_3 x^2 + \dots + a_{n-1} x^{n-2})$$

i supstitucijom $x' = x^2$.

Dakle, definiramo "parni" i "neparni" polinom

$$A^{[0]}(x) = a_0 + a_2 x + \dots + a_{n-2} x^{n/2-1}$$
$$A^{[1]}(x) = a_1 + a_3 x + \dots + a_{n-1} x^{n/2-1}$$

pa je
$$A(x) = A^{[0]}(x^2) + x \cdot A^{[1]}(x^2).$$

Označimo s $a^{[0]}$ i $a^{[1]}$ pripadne nizove koeficijenata

$$a^{[0]} = (a_0, a_2, -, a_{n-2})$$
$$a^{[1]} = (a_1, a_3, -, a_{n-1}).$$

Vidimo da $a^{[0]}$ sadrži one koeficijente čiji indeks ima zadnju znamenku jednaku 0 u bazi 2, a $a^{[1]}$ one čiji indeks ima zadnju znamenku 1 u bazi 2.

Broj ili baza 2 je faktor cijepanja ili rastava

$$\underbrace{n}_{\text{stari red}} = \underbrace{2}_{\text{faktor}} \cdot \underbrace{\left(\frac{n}{2}\right)}_{\text{novi red}}.$$

Neka su $y^{[0]}$ i $y^{[1]}$ pripadne diskretne Fourierove transformacije polovnog reda

$$y^{[0]} := DFT_{n/2}(a^{[0]})$$
$$y^{[1]} := DFT_{n/2}(a^{[1]}).$$

"Puni" DFT polaznog niza koeficijenata

$$y = DFT_n(a)$$

dobivamo pažljivom kombinacijom "polovica" vektora $y^{[0]}$ i $y^{[1]}$

prva polovica: $\quad y_k = y_k^{[\emptyset]} + \omega_n^k \cdot y_k^{[1]}$

druga polovica:

$$y_{k+n/2} = y_k^{[\emptyset]} + \omega_n^{k+n/2} \cdot y_k^{[1]}$$
$$= y_k^{[\emptyset]} - \omega_n^{k} \cdot y_k^{[1]}$$

za $k = 0, 1, \ldots, n/2$.

Vidimo da je drugi član isti, do na predznak, pa odmah možemo uštedjeti jedno množenje, ako izračunamo pomoćnu vrijednost

$$t = \omega_n^k \cdot y_k^{[1]}$$

a zatim

$$y_k = y_k^{[\emptyset]} + t, \quad y_{k+n/2} = y_k^{[\emptyset]} - t .$$

Odgovarajući rekurzivni algoritam, pisan funkcijski (nalik na Matlab ili C) ima oblik:

```
function FFT (a);   { a je kompleksni vektor,
                      vraća y = DFT_n(a) }
  n := length(a);   { pretpostavka je n = 2^m, m ∈ N_0 ! }
  if n = 1 then
     return a          { y = a za n = 1 }
  else
     a[∅] := (a_0, a_2, ..., a_{n-2});
     a[1] := (a_1, a_3, ..., a_{n-1});
     y[∅] := FFT(a[∅]);
     y[1] := FFT(a[1]);
     ω_n := e^{2πi/n}; {ω := 1;}
     for k := ∅ to n/2-1 do                      n/2 − 1
        t := ω_n^k · y_k[1];     { = ω · y_k[1] }
        y_k := y_k[∅] + t;
        y_{k+n/2} := y_k[0] − t;
        { ω := ω * ω_n }
     end for;
  return y;        { kompleksni vektor duljine n }
```

Skaliranje, ako treba, napravimo <u>nakon</u> povratka!

[Ovdje dobite složenost $\left\{ \begin{array}{l} \frac{1}{2} n \lg n \ M \\ n \lg n \ A \end{array} \right\}$ uz table-lookup za ω_n^k !]

$n = 2^m$ za $m = \emptyset$ $T(1) = \emptyset$

općenito at stage $n = 2^m$ - imam for od $\emptyset$ do $n/2 - 1$

tj. $2^{m-1} \times$:

$$\left.\begin{array}{l} 1A + 1M \\ 1A + 1M \end{array}\right\} \quad 1M + 2A$$

$$1M \text{ ——— } 1M$$

$$\overline{2M + 2A}$$

ili $2^m \cdot M + 2^m \cdot A$

Dakle: $M(n) = 2 \cdot M(n/2) + n$ i isto za A

$M(n/2) = 2 \cdot M(n/4) + n/2$

$$\Rightarrow \quad M(n) = 4 \cdot M(n/4) + n + n$$
$$= 4 \cdot \left(2M\left(\tfrac{n}{8}\right) + \tfrac{n}{4}\right) + n + n$$
$$= 8 \cdot M\left(\tfrac{n}{8}\right) + 3 \cdot n$$

Općenito: $M(n) = 2^k \cdot M\left(\frac{n}{2^k}\right) + k \cdot n$

$$= 2^k \cdot \left(2 \cdot M\left(\tfrac{n}{2^{k+1}}\right) + \tfrac{n}{2^k}\right) + k \cdot n$$
$$= 2^{k+1} M\left(\tfrac{n}{2^{k+1}}\right) + (k+1) \cdot n$$

za $k = \lg n$ $M(n) = n \cdot \underbrace{M(1)}_{0} + n \cdot \lg n = n \cdot m$

Dakle, broj kompl. zbrajanja $\boxed{A(n) = n \cdot \lg n}$

- Ako spremim pomoćno: $\omega \cdot y_k^{[1]} \rightarrow \boxed{M_1(n) = \frac{1}{2} n \cdot \lg n}$

a oper. $\underbrace{\omega := \omega \cdot \omega_n}$ košta još $M_2(n) = \frac{1}{2} n \cdot \lg n$

ovo mogu tabelirati i spremiti
u polje duljine n za $M_2(n) = n - 1$

- Par FFT, FFT^{-1} $\rightarrow$ treba $2\times$ i dodati (barem) $n M$
za final scaling
(ili $2n$ M, ako oba imaju const
na pr. $\frac{1}{\sqrt{n}}$)

Inverzna diskretna Fourierova transformacija DFT_n^{-1} je naravno

$$a = DFT_n^{-1}(y) \iff a = V_n^{-1} \cdot y .$$

Prošli puta smo vidjeli da je $V_n = V_n^T$ i da za inverz vrijedi

$$V_n^{-1} = \frac{1}{n} \cdot V_n^* .$$

Zbog $\overline{\omega}_n = \omega_n^{-1}$, DFT_n^{-1} dobivamo tako da u DFT_n, umjesto ω_n, koristimo $\omega_n^{-1} = \overline{\omega}_n$ i, na kraju skaliramo finalni vektor s $1/n$.

U algoritamskim implementacijama se vrlo često ignorira ovo skaliranje na kraju i dodaje samo kad je nužno potrebno i to na samom kraju svih transformacija.

Zbog toga, to skaliranje nećemo posebno brojati u analizi složenosti, opet, osim u kompliciranijim algoritmima — s više DFT, DFT^{-1} transformacija, kad je to skaliranje bitno za korektan konačni rezultat.

- Nekoliko komentara, zbog raznih oznaka i imena u literaturi.

1. U teoriji je najugodnije raditi s matricom
$$U_n = \frac{1}{\sqrt{n}} \cdot V_n$$
(simetrična skala za DFT_n i DFT_n^{-1} — oba imaju <u>istu</u> skalu), zato što je
$$U_n^{-1} = U_n^* ,$$
pa je U_n unitarna, što je zgodno za razne stvari u teoriji (čuva skalarne produkte i norme vektora!).

Tada se DFT_n^{-1} dobiva iz DFT_n samo zamjenom
$$\underset{(DFT_n)}{\omega_n} \longmapsto \underset{(DFT_n^{-1})}{\omega_n^{-1} = \overline{\omega}_n} .$$