Szybka transformacja Fouriera (FFT)

Problem:

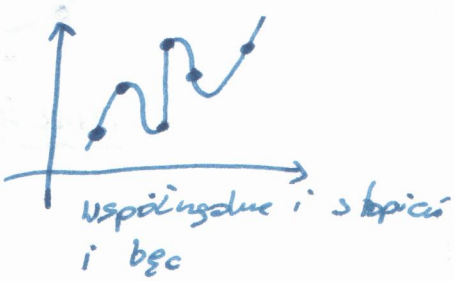
Dane: $a_0, a_1, \ldots, a_{n-1}$
$b_0, b_1, \ldots, b_{n-1}$
> interpr. współczynniki $\begin{cases} A(x) \\ B(x) \end{cases}$

Wynik: $c_0, c_1, \ldots, c_{2n-2}$ – inter. współczynniki $C(x) = A(x)B(x)$

$$\forall_{0 \leq i \leq 2n-2} \quad c_i = \sum_{j=0}^{i} a_j b_{i-j}$$

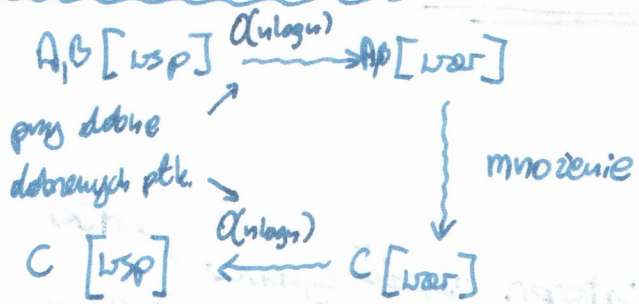
Naiwnie: $\Theta(n^2)$

2 reprezentacje wielomianów:
- [wsp] – ciąg współczynników
- [war] – zbiór wartości w punktach



Operacje na wielomianach:
- obliczenie wartości dla danego x
  - [wsp] – $O(n)$
  - [war] – prykie
- dodawanie wielomianów:
  - [wsp] – $O(n)$
  - [war] – $O(n)$
- mnożenie wielomianów
  - [wsp] – naiwnie $\Theta(n^2)$
  - [war] – $O(n)$ ← to jest lekkie oszustwo

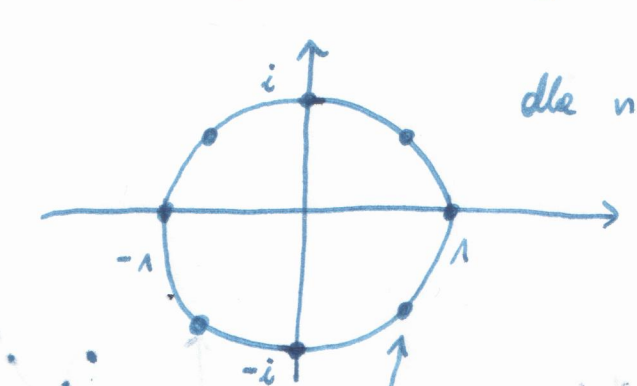
Schemat algorytmu:



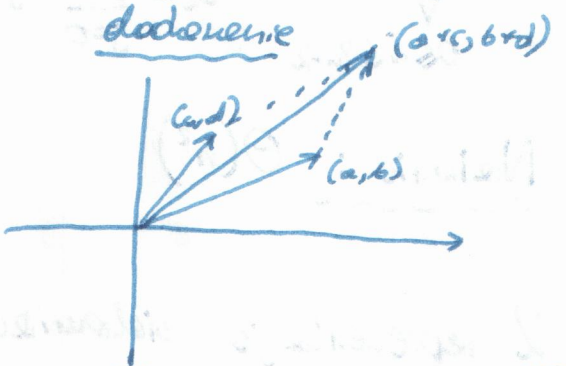
Jak wybrać punkty do reprezentacji [war] ?

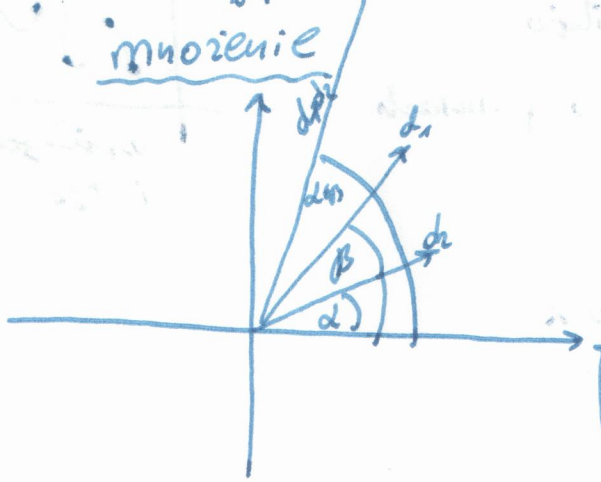
(bo jak zrobimy : wybierzemy źle, to będzie $O(n^2)$

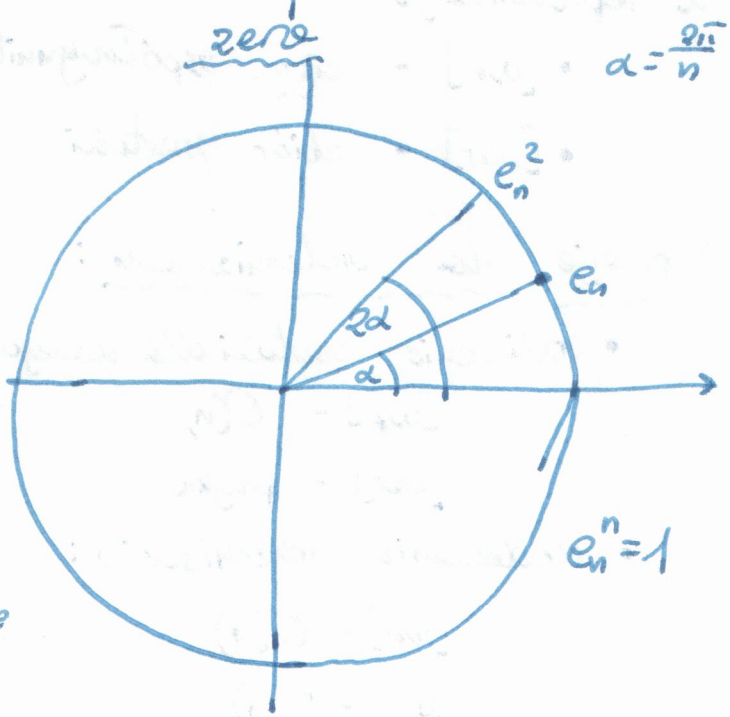
- n-te pierwiastki z jedności $\{x : x^n = 1\}$



dla $n=8$







e_n - jest pierwiastkiem pierwotnym, bo generuje wszystkie inne

FAKT Niech n-parzyste $\{e_n^{2\cdot 0}, e_n^{2\cdot 1}, e_n^{2\cdot 2}, e_n^{2\cdot 3}, \ldots, e_n^{2(n-1)}\} =$

$= \{e_{n/2}^0, e_{n/2}^1, e_{n/2}^2, e_{n/2}^3, \ldots, e_{n/2}^{\frac{n}{2}-1}\}$

Zał. $n = 2^k$

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

Tworzymy 2 wielom.

$$A_p(z) = a_0 + a_2 z + a_4 z^2 + a_6 z^3 + \ldots + a_{n-2} z^{n/2-1}$$

$$A_n(z) = a_1 + a_3 z + a_5 z^2 + a_7 z^3 + \ldots a_{n-1} z^{n/2-1}$$

$$A(x) = A_p(x^2) + x A_n(x^2)$$

chcąc obliczyć wartości $A(x)$ dla

$$x = \omega_n^0, \omega_n^1, \omega_n^2, \ldots, \omega_n^{n-1}$$

obliczmy wartości $A_p(z)$ i $A_n(z)$ dla

$z = (\omega_n^0)^2, (\omega_n^1)^2, (\omega_n^2)^2, \ldots, (\omega_n^{n-1})^2$, czyli z faktu wiemy że

$z = \omega_{n/2}^0, \omega_{n/2}^1, \omega_{n/2}^2, \ldots, \omega_{n/2}^{n/2-1}$

Niech $\bar{a} = a_0, a_1, a_2, \ldots, a_{n-1}$

```
rec FFT(ā)
    if (n=1) return ā
    ω_n ← e^(2πi/n)
    ω ← 1
    ā^[0] ← ⟨a_0, a_2, a_4, ..., a_{n-2}⟩
    ā^[1] ← ⟨a_1, a_3, a_5, ..., a_{n-1}⟩
    ȳ^[0] ← rec FFT(ā^[0])
    ȳ^[1] ← rec FFT(ā^[1])
    for k ← 0 ... n/2 - 1
        y_k ← ȳ_k^[0] + ω ȳ_k^[1]
        y_{k+n/2} ← ȳ_k^[0] - ω ȳ_k^[1]
        ω ← ω ω_n
    return ⟨y_0, y_1, ..., y_{n-1}⟩
```

Czas $T(n) = 2T(\frac{n}{2}) + O(n)$,

czyli $T(n) = O(n \log n)$

Jak przejść z [wzr] do [wsp]?

Na wejściu dla danego $\bar{a} = \langle a_0, \dots, a_{n-1} \rangle$ obliczamy $\bar{y} = \langle y_0, \dots, y_{n-1} \rangle$, tzn $y_k = \sum_{j=0}^{n-1} a_j (\omega_n^k)^j = \sum a_j \omega_n^{kj}$

$$\bar{y} = V_n \bar{a}$$

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} \omega_n^{00} & \omega_n^{01} & \omega_n^{02} & \dots & \omega_n^{0(n-1)} \\ \omega_n^{10} & \omega_n^{11} & \omega_n^{12} & \dots & \omega_n^{1(n-1)} \\ \vdots & & & & \vdots \\ \omega_n^{(n-1)0} & \omega_n^{(n-1)1} & \omega_n^{(n-1)2} & \dots & \omega_n^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

W i-tym wierszu i j-tej kolumnie macierzy V_n jest wartość $\omega_n^{i \cdot j}$. Teraz mamy $\bar{y}$ i chcemy obliczyć $\bar{a}$

$$\bar{a} = V_n^{-1} \bar{y}$$

Macierz V_n jest odwracalna. W i-tym wierszu i j-tej kolumnie macierzy V_n^{-1} jest wartość $\frac{1}{n} \omega_n^{-ij}$. Po wyciągnięciu $\frac{1}{n}$ przed macierz na i,j-tym miejscu jest $(\omega_n^{-1})^{ij}$ ← i to też jest n-ty pierwotny pierwiastek z jedności. Czyli obliczenie $\frac{1}{n} V_n^{-1} \cdot \bar{y}$, a to można uzyskać poprzez FFT.

Jeśli $a_0, \dots, a_{n-1}, b_0, \dots, b_{n-1} \in \mathbb{N}$. Obliczanie możemy wykonywać w $\mathbb{Z}_p$, $p \in \mathbb{P}$, gdzie p jest odpowiednio duże i w $\mathbb{Z}_p$ jest n-ty pierwotny pierwiastek z jedności, który ma własność:

$$(*) \quad \forall_{\substack{k \geq 0 \\ n \nmid k}} \sum_{j=0}^{n-1} (\omega_n^k)^j = 0$$

FAKT: Niech n, w - potęgi liczby 2 ($\neq 1$)

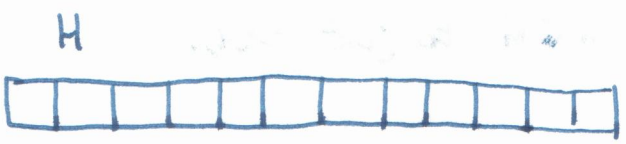
$$m = \omega^{n/2} + 1$$

Wówczas n i w są odwracalne w pierścieniu reszt mod m oraz ω jest n-tym pierwiastkiem pierwotnym z jedności spełniającym $(*)$.

Haszowanie

U - universum kluczy, $S \subseteq U$

operacje pożądane:
- insert
- delete
- search



$H[i]$ - true iff $i \in S$

nierealne, gdy $|U|$ - wielkie

$m = O(|S|)$
$\uparrow$
n

Chcemy mieć funkcję $h: U \to \{0, \dots, m-1\}$
↖ f-ga haszująca

Kolizja - sytuacja, w której $\forall_{\substack{k,l \in S \\ k \neq l}} h(k) = h(l)$