Transformada rápida de Fourier

Parte III

Segundo semestre 2022

IIC2283

Prof. Nicolás Van Sint Jan

Recordatorio: Multiplicación de polinomios

$$
\begin{tikzcd}
\boxed{\begin{array}{l} p(x) \mapsto (a_0, \ldots, a_{n-1}) \\ q(x) \mapsto (b_0, \ldots, b_{n-1}) \end{array}} \arrow[r, "O(n^2)"] \arrow[d, "O(n^2)"'] & \boxed{\begin{array}{l} p(x) \mapsto \{(v_0, p(v_0)), \ldots, (v_{2n-1}, p(v_{2n-1}))\} \\ q(x) \mapsto \{(v_0, q(v_0)), \ldots, (v_{2n-1}, q(v_{2n-1}))\} \end{array}} \arrow[d, "O(n)"] \\
\boxed{p(x) \cdot q(x) \mapsto (c_0, \ldots, c_{2n-2})} & \boxed{\begin{array}{r} p(x) \cdot q(x) \mapsto \{(v_0, p(v_0) \cdot q(v_0)), \ldots \\ \ldots, (v_{2n-1}, p(v_{2n-1}) \cdot q(v_{2n-1}))\} \end{array}} \arrow[l, "O(n^2)"]
\end{tikzcd}
$$

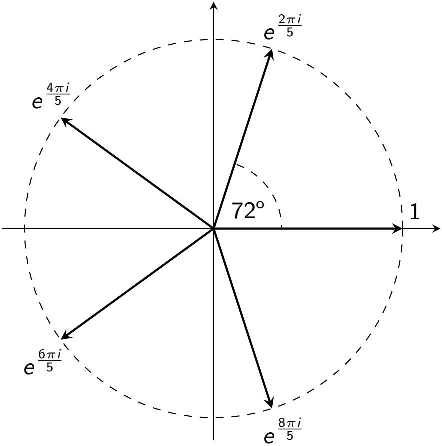
Recordatorio: Transformada rápida de Fourier

$$
\begin{tikzcd}[column sep=large, row sep=huge]
\boxed{\begin{array}{l} p(x) \mapsto (a_0, \ldots, a_{n-1}) \\ q(x) \mapsto (b_0, \ldots, b_{n-1}) \end{array}} \arrow[r, red, "O(n \cdot \log_2(n))"] \arrow[d, "O(n^2)"'] & \boxed{\begin{array}{l} p(x) \mapsto \{(v_0, p(v_0)), \ldots, (v_{2n-1}, p(v_{2n-1}))\} \\ q(x) \mapsto \{(v_0, q(v_0)), \ldots, (v_{2n-1}, q(v_{2n-1}))\} \end{array}} \arrow[d, "O(n)"] \\
\boxed{p(x) \cdot q(x) \mapsto (c_0, \ldots, c_{2n-2})} & \boxed{\begin{array}{r} p(x) \cdot q(x) \mapsto \{(v_0, p(v_0) \cdot q(v_0)), \ldots \\ \ldots, (v_{2n-1}, p(v_{2n-1}) \cdot q(v_{2n-1}))\} \end{array}} \arrow[l, red, "O(n \cdot \log_2(n))"]
\end{tikzcd}
$$

Recordatorio: Raíces de la unidad

Proposición

Si $\omega_n = e^{\frac{2\pi i}{n}}$, las n-raíces de la unidad son ω_n^0, ω_n^1, ω_n^2, ..., ω_n^{n-1}



Recordatorio: La transformada discreta de Fourier

Definición

Sea $n \geq 2$ y un polinomio $p(x) = \sum_{k=0}^{n-1} a_k x^k$.

La **transformada discreta de Fourier (DFT)** de $p(x)$ se define como:

$$[p(\omega_n^0), p(\omega_n^1), \ldots, p(\omega_n^{n-1})]$$

Recordatorio: La descomposición recursiva

Tenemos que:

$$\begin{aligned} p(x) &= \sum_{k=0}^{n-1} a_k x^k \\ &= \sum_{k=0}^{\frac{n}{2}-1} a_{2k}\ x^{2k} + \sum_{k=0}^{\frac{n}{2}-1} a_{2k+1}\ x^{2k+1} \\ &= \sum_{k=0}^{\frac{n}{2}-1} a_{2k}\ x^{2k} + x \cdot \sum_{k=0}^{\frac{n}{2}-1} a_{2k+1}\ x^{2k} \end{aligned}$$

Definimos los polinomios:

$$\begin{aligned} q(z) &= \sum_{k=0}^{\frac{n}{2}-1} a_{2k}\ z^k \\ r(z) &= \sum_{k=0}^{\frac{n}{2}-1} a_{2k+1}\ z^k \end{aligned}$$

Recordatorio: La descomposición recursiva

Tenemos entonces que

$$p(x) = q(x^2) + x \cdot r(x^2)$$

Si $k \in \{0, \ldots, \frac{n}{2} - 1\}$, entonces tenemos que:

$$\begin{aligned} (\omega_n^{\frac{n}{2}+k})^2 &= \omega_n^{n+2k} \\ &= \omega_n^n \cdot \omega_n^{2k} \\ &= 1 \cdot (\omega_n^k)^2 \\ &= (\omega_n^k)^2 \end{aligned}$$

Por lo tanto para calcular $[p(\omega_n^0), p(\omega_n^1), \ldots, p(\omega_n^{n-1})]$, basta con calcular:

$$[q((\omega_n^0)^2), q((\omega_n^1)^2), \ldots, q((\omega_n^{\frac{n}{2}-1})^2)]$$
$$[r((\omega_n^0)^2), r((\omega_n^1)^2), \ldots, r((\omega_n^{\frac{n}{2}-1})^2)]$$

Recordatorio: Las $\frac{n}{2}$-raíces de la unidad

Lema

Si $n \geq 2$ es par, entonces $(\omega_n^0)^2$, $(\omega_n^1)^2$, $\ldots$, $(\omega_n^{\frac{n}{2}-1})^2$ son las $\frac{n}{2}$-raíces de la unidad (vale decir, son la raíces el polinomio $x^{\frac{n}{2}} - 1$).

Outline

DFT (cont.)

La inversa de la DFT

Outline

DFT (cont.)

La inversa de la DFT

La descomposición recursiva (continuación)

Recuerde que $\bar{a} = (a_0, \ldots, a_{n-1})$, y defina:

$$\begin{aligned} \bar{a}_0 &= (a_0, a_2, \ldots, a_{n-2}) \\ \bar{a}_1 &= (a_1, a_3, \ldots, a_{n-1}) \end{aligned}$$

De los resultados anteriores concluimos que para calcular $\mathbf{DFT}(\bar{a})$, primero tenemos que calcular $\mathbf{DFT}(\bar{a}_0)$ y $\mathbf{DFT}(\bar{a}_1)$

¿ Cómo se construye $\mathbf{DFT}(\bar{a})$ a partir de $\mathbf{DFT}(\bar{a}_0)$ y $\mathbf{DFT}(\bar{a}_1)$?

Construyendo $\mathbf{DFT}(\bar{a})$ a partir de $\mathbf{DFT}(\bar{a}_0)$ y $\mathbf{DFT}(\bar{a}_1)$

Sea:

$$\begin{aligned}
\mathbf{DFT}(\bar{a}_0) &= [u_0, u_1, \ldots, u_{\frac{n}{2}-1}] \\
\mathbf{DFT}(\bar{a}_1) &= [v_0, v_1, \ldots, v_{\frac{n}{2}-1}]
\end{aligned}$$

Para $k \in \{0, \ldots, \frac{n}{2} - 1\}$ tenemos que:

$$\begin{aligned}
y_k &= p(\omega_n^k) \\
&= q((\omega_n^k)^2) + \omega_n^k \cdot r((\omega_n^k)^2) \\
&= q(\omega_{\frac{n}{2}}^k) + \omega_n^k \cdot r(\omega_{\frac{n}{2}}^k) \\
&= u_k + \omega_n^k \cdot v_k
\end{aligned}$$

Construyendo $\mathbf{DFT}(\bar{a})$ a partir de $\mathbf{DFT}(\bar{a}_0)$ y $\mathbf{DFT}(\bar{a}_1)$

Además, para $k \in \{0, \ldots, \frac{n}{2} - 1\}$ tenemos que:

$$\begin{aligned}
y_{\frac{n}{2}+k} &= p(\omega_n^{\frac{n}{2}+k}) \\
&= q((\omega_n^{\frac{n}{2}+k})^2) + \omega_n^{\frac{n}{2}+k} \cdot r((\omega_n^{\frac{n}{2}+k})^2) \\
&= q(\omega_n^{n+2k}) + \omega_n^{\frac{n}{2}} \cdot \omega_n^k \cdot r(\omega_n^{n+2k}) \\
&= q(\omega_n^n \cdot \omega_n^{2k}) + (e^{\frac{2\pi i}{n}})^{\frac{n}{2}} \cdot \omega_n^k \cdot r(\omega_n^n \cdot \omega_n^{2k}) \\
&= q(1 \cdot \omega_n^{2k}) + e^{\pi i} \cdot \omega_n^k \cdot r(1 \cdot \omega_n^{2k}) \\
&= q(\omega_n^{2k}) - \omega_n^k \cdot r(\omega_n^{2k}) \\
&= q(\omega_{\frac{n}{2}}^k) - \omega_n^k \cdot r(\omega_{\frac{n}{2}}^k) \\
&= u_k - \omega_n^k \cdot v_k
\end{aligned}$$

Construyendo $\mathbf{DFT}(\bar{a})$ a partir de $\mathbf{DFT}(\bar{a}_0)$ y $\mathbf{DFT}(\bar{a}_1)$

Resumiendo, para $k \in \{0, \ldots, \frac{n}{2} - 1\}$ tenemos que:

$$\begin{aligned} y_k &= u_k + \omega_n^k \cdot v_k \\ y_{\frac{n}{2}+k} &= u_k - \omega_n^k \cdot v_k \end{aligned}$$

Para tener un algoritmo recursivo para calcular **DFT** sólo nos falta el **caso base**.

- Consideramos $n = 2$ y un polinomio $p(x) = a_0 + a_1 x$

Construyendo $\mathbf{DFT}(\bar{a})$ a partir de $\mathbf{DFT}(\bar{a}_0)$ y $\mathbf{DFT}(\bar{a}_1)$

Tenemos que

$$
\begin{aligned}
p(\omega_2^0) &= a_0 + a_1 \cdot \omega_2^0 = a_0 + a_1 \\
p(\omega_2^1) &= a_0 + a_1 \cdot \omega_2^1 = a_0 - a_1
\end{aligned}
$$

Un algoritmo recursivo eficiente para **DFT**

- La entrada del algoritmo es un polinomio $p(x) = \sum_{k=0}^{n-1} a_k x^k$
  - Este polinomio es representado por el vector $\bar{a} = (a_0, \ldots, a_{n-1})$

- Suponemos además que $n \geq 2$ y n es una potencia de 2.

- El algoritmo se llama la **transformada rápida de Fourier (FFT)**.
  - Fue propuesto por Cooley & Tukey (1965).

Un algoritmo recursivo eficiente para **DFT**

FFT$(a_0, \ldots, a_{n-1})$

  if $n = 2$ **then**

    $y_0 = a_0 + a_1$

    $y_1 = a_0 - a_1$

    return $[y_0, y_1]$

  else

    $[u_0, \ldots, u_{\frac{n}{2}-1}] :=$ **FFT**$(a_0, \ldots, a_{n-2})$

    $[v_0, \ldots, v_{\frac{n}{2}-1}] :=$ **FFT**$(a_1, \ldots, a_{n-1})$

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

    $\alpha := 1$

    for $k := 0$ **to** $\frac{n}{2} - 1$ **do**

      $y_k := u_k + \alpha \cdot v_k$

      $y_{\frac{n}{2}+k} := u_k - \alpha \cdot v_k$

      $\alpha := \alpha \cdot \omega_n$

    return $[y_0, \ldots, y_{n-1}]$

Algunos comentarios sobre **FFT**

Ejercicios

1. Demuestre que **FFT** funciona en tiempo $O(n \cdot \log_2(n))$, donde n es el grado del polinomio de entrada.

2. Sea $p(x)$ un polinomio representado como $\bar{a} = (a_0, \ldots, a_{n-1})$, donde n **no** es una potencia de 2.

   Para evaluar $p(x)$ en n puntos haga lo siguiente:

   (i) Defina $\bar{b} = (a_0, \ldots, a_{n-1}, 0, \ldots, 0)$ de largo $m = 2^{\lceil \log_2(n) \rceil}$

   (ii) Calcule **FFT**$(\bar{b}) = [y_0, \ldots, y_{m-1}]$, y retorne $[y_0, \ldots, y_{n-1}]$

   Note que este algoritmo retorna $[p(\omega_m^0), \ldots, p(\omega_m^{n-1})]$ en lugar de **DFT**$(\bar{a})$, y demuestre que funciona en tiempo $O(n \cdot \log_2(n))$

La nueva situación con el algoritmo **FFT**

$$\bar{a} = \overbrace{(a_0, \ldots, a_{n-1}, 0, \ldots, 0)}^{\text{largo } 2n}$$
$$\bar{b} = (b_0, \ldots, b_{n-1}, 0, \ldots, 0)$$
$$p(x) \mapsto (a_0, \ldots, a_{n-1})$$
$$q(x) \mapsto (b_0, \ldots, b_{n-1})$$

$$\xrightarrow{O(n \cdot \log_2(n))}$$

$$\textbf{FFT}(\bar{a}) = [y_0, \ldots, y_{2n-1}]$$
$$\textbf{FFT}(\bar{b}) = [z_0, \ldots, z_{2n-1}]$$
$$p(x) \mapsto \{(\omega_{2n}^0, y_0), \ldots, (\omega_{2n}^{2n-1}, y_{2n-1})\}$$
$$q(x) \mapsto \{(\omega_{2n}^0, z_0), \ldots, (\omega_{2n}^{2n-1}, z_{2n-1})\}$$

$$\Big\downarrow O(n)$$

$$p(x) \cdot q(x) \mapsto \{(\omega_{2n}^0, y_0 \cdot z_0), \ldots$$
$$\ldots, (\omega_{2n}^{2n-1}, y_{2n-1} \cdot z_{2n-1})\}$$

$$p(x) \cdot q(x) \mapsto (c_0, \ldots, c_{2n-2})$$

Todavía nos falta un algoritmo para calcular la inversa de la transformada discreta de Fourier.

Outline

DFT (cont.)

La inversa de la DFT

La transformada discreta de Fourier como matriz

Suponga nuevamente que $p(x) = \sum_{k=0}^{n-1} a_k x^k$ con $n \geq 2$. Luego para cada $k \in \{0, \ldots, n-1\}$:

$$y_k = p(\omega_n^k) = \sum_{i=0}^{n-1} a_i (\omega_n^k)^i = \sum_{i=0}^{n-1} a_i \cdot \omega_n^{i \cdot k}$$

La transformada discreta de Fourier puede ser representada entonces de la siguiente forma en **notación matricial**:

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)(n-1)} \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

La transformada discreta de Fourier como matriz

Definimos la ***n*-ésima matriz de Fourier $\mathbf{F}_n$** como:

$$\mathbf{F}_n = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)(n-1)} \end{pmatrix}$$

Esta matriz va a ser muy útil al momento de definir la **inversa de la transformada discreta de Fourier**.

La matrices $\mathbf{F}_n$ y $\mathbf{F}_n^*$

Para cada $i, j \in \{1, \ldots, n\}$, tenemos que

$$\mathbf{F}_n[i,j] = (\omega_n^{i-1})^{j-1} = \omega_n^{(i-1)\cdot(j-1)}$$

Por lo tanto $\mathbf{F}_n[i,j] = \mathbf{F}_n[j,i]$, de lo cual se deduce que $\mathbf{F}_n$ es una **matriz simétrica**.

Definición

La **matriz adjunta** A^* de una matriz A se define como

$$A^*[i,j] = \overline{A[j,i]}$$

Donde $\overline{b + ci} = b - ci$ es el conjugado del número complejo $b + ci$

Para calcular $\mathbf{F}_n^*$ necesitamos calcular el conjugado de ω_n^k

El conjugado de ω_n^k

Tenemos que:

$$
\begin{array}{rcl}
\overline{(\omega_n^k)} & = & \overline{\left(e^{\frac{2\pi i}{n}}\right)^k} \\
& = & \overline{e^{\frac{2\pi ki}{n}}} \\
& = & \overline{\cos\left(\frac{2\pi k}{n}\right) + i\sin\left(\frac{2\pi k}{n}\right)} \\
& = & \cos\left(\frac{2\pi k}{n}\right) - i\sin\left(\frac{2\pi k}{n}\right) \\
& = & \cos\left(-\frac{2\pi k}{n}\right) + i\sin\left(-\frac{2\pi k}{n}\right) \\
& = & e^{\frac{-2\pi ki}{n}} \\
& = & (e^{\frac{2\pi i}{n}})^{-k} \\
& = & \omega_n^{-k}
\end{array}
$$

Las matrices $\mathbf{F}_n$ y $\mathbf{F}_n^*$ (continuación)

Tenemos entonces que para $i, j \in \{1, \ldots, n\}$:

$$\begin{aligned} \mathbf{F}_n^*[i,j] &= \overline{\mathbf{F}_n[j,i]} \\ &= \overline{(\omega_n^{j-1})^{i-1}} \\ &= \overline{\omega_n^{(i-1)\cdot(j-1)}} \\ &= \omega_n^{-(i-1)\cdot(j-1)} \end{aligned}$$

Vamos a utilizar esta propiedad para calcular $\mathbf{F}_n^* \cdot \mathbf{F}_n$.

Las matrices $\mathbf{F}_n$ y $\mathbf{F}_n^*$ (continuación)

Lema

Sea I_n la matriz identidad de tamaño n. Entonces $\mathbf{F}_n^* \cdot \mathbf{F}_n = n \cdot I_n$.

Demostración

Sea $A = \mathbf{F}_n^* \cdot \mathbf{F}_n$.

Para $i \in \{1, \ldots, n\}$, tenemos que:

$$\begin{aligned}
A[i,i] &= \sum_{k=1}^{n} M_n^*[i,k] \cdot \mathbf{F}_n[k,i] \\
&= \sum_{k=1}^{n} \omega_n^{-(i-1)\cdot(k-1)} \cdot \omega_n^{(k-1)\cdot(i-1)} \\
&= \sum_{k=1}^{n} 1 \\
&= n
\end{aligned}$$

Las matrices $\mathbf{F}_n$ y $\mathbf{F}_n^*$ (continuación)

Demostración

Para $i, j \in \{1, \ldots, n\}$ con $i \neq j$, tenemos que:

$$\begin{aligned}
A[i,j] &= \sum_{k=1}^{n} \mathbf{F}_n^*[i,k] \cdot \mathbf{F}_n[k,j] \\
&= \sum_{k=1}^{n} \omega_n^{-(i-1)\cdot(k-1)} \cdot \omega_n^{(k-1)\cdot(j-1)} \\
&= \sum_{k=1}^{n} \left(\omega_n^{-(i-1)+(j-1)}\right)^{k-1} \\
&= \sum_{k=1}^{n} (\omega_n^{j-i})^{k-1} \\
&= \frac{(\omega_n^{j-i})^n - 1}{\omega_n^{j-i} - 1} \qquad \text{(¿Por qué?)} \\
&= \frac{(\omega_n^n)^{j-i} - 1}{\omega_n^{j-i} - 1} = \frac{1^{j-i} - 1}{\omega_n^{j-i} - 1} = \frac{1 - 1}{\omega_n^{j-i} - 1} = 0
\end{aligned}$$

$\square$

$\mathbf{F}_n^*$ como una permutación de $\mathbf{F}_n$

Dado $i \in \{1, \ldots, n\}$ y $j \in \{2, \ldots, n\}$, tenemos que:

$$
\begin{aligned}
\mathbf{F}_n[i, n+2-j] &= \omega_n^{(i-1)\cdot(n+2-j-1)} \\
&= \omega_n^{(i-1)\cdot(n+1-j)} \\
&= \omega_n^{n\cdot(i-1)+(i-1)\cdot(1-j)} \\
&= \omega_n^{n\cdot(i-1)} \cdot \omega_n^{(i-1)\cdot(1-j)} \\
&= (\omega_n^n)^{i-1} \cdot \omega_n^{-(i-1)\cdot(j-1)} \\
&= 1^{i-1} \cdot \omega_n^{-(i-1)\cdot(j-1)} \\
&= \omega_n^{-(i-1)\cdot(j-1)} \\
&= \mathbf{F}_n^*[i, j]
\end{aligned}
$$

$\mathbf{F}_n^*$ como una permutación de $\mathbf{F}_n$

Para $j \in \{2, \ldots, n\}$, concluimos que:

$$\begin{pmatrix} \mathbf{F}_n^*[1,j] \\ \mathbf{F}_n^*[2,j] \\ \vdots \\ \mathbf{F}_n^*[n,j] \end{pmatrix} = \begin{pmatrix} \mathbf{F}_n[1, n+2-j] \\ \mathbf{F}_n[2, n+2-j] \\ \vdots \\ \mathbf{F}_n[n, n+2-j] \end{pmatrix}$$

Vamos a usar esta propiedad para definir $\mathbf{F}_n^*$ como una **permutación** de $\mathbf{F}_n$.

$\mathbf{F}_n^*$ como una permutación de $\mathbf{F}_n$

Considere la siguiente matriz de permutación de $n \times n$:

$$P_n \quad = \quad \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \iddots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

De la propiedad en la transparencia anterior se concluye que:

$$\mathbf{F}_n^* \quad = \quad \mathbf{F}_n \cdot P_n$$

Calculando la inversa de la DFT

Teorema

Sea $n \geq 2$, $\bar{a} = (a_0, a_1, \ldots, a_{n-1})$ y $\mathbf{DFT}(\bar{a}) = [y_0, y_1, \ldots, y_{n-1}]$.

Si $\bar{p} = (y_0, y_{n-1}, \ldots, y_1)$, entonces:

$$\frac{1}{n} \cdot \mathbf{DFT}(\bar{p}) \quad = \quad [a_0, a_1, \ldots, a_{n-1}]$$

La demostración del teorema

Demostración

Tenemos que:

$$\mathbf{F}_n \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

Por lo tanto:

$$P_n \cdot \mathbf{F}_n \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = P_n \cdot \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_{n-1} \\ \vdots \\ y_1 \end{pmatrix}$$

La demostración del teorema

Demostración

Concluimos que:

$$\mathbf{F}_n \cdot P_n \cdot \mathbf{F}_n \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} \;=\; \mathbf{F}_n \cdot \begin{pmatrix} y_0 \\ y_{n-1} \\ \vdots \\ y_1 \end{pmatrix}$$

Dado que $\mathbf{F}_n \cdot P_n = \mathbf{F}_n^*$, tenemos que:

$$M_n^* \cdot \mathbf{F}_n \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} \;=\; \mathbf{F}_n \cdot \begin{pmatrix} y_0 \\ y_{n-1} \\ \vdots \\ y_1 \end{pmatrix}$$

La demostración del teorema

Demostración

Así, puesto que $\mathbf{F}_n^* \cdot \mathbf{F}_n = n \cdot I_n$, tenemos que:

$$n \cdot I_n \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} \;=\; \mathbf{F}_n \cdot \begin{pmatrix} y_0 \\ y_{n-1} \\ \vdots \\ y_1 \end{pmatrix}$$

De lo cual concluimos que:

$$\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} \;=\; \frac{1}{n} \cdot \mathbf{F}_n \cdot \begin{pmatrix} y_0 \\ y_{n-1} \\ \vdots \\ y_1 \end{pmatrix}$$

La demostración del teorema

Demostración

Así, utilizando la notación matricial para la transformada discreta de Fourier concluimos que:

$$[a_0, a_1, \ldots, a_{n-1}] \quad = \quad \frac{1}{n} \cdot \mathbf{DFT}(\bar{p})$$

$\square$

Multiplicando polinomios en tiempo $O(n \cdot \log n)$

$$\bar{a} = \overbrace{(a_0, \dots, a_{n-1}, 0, \dots, 0)}^{\text{largo } 2n}$$
$$\bar{b} = (b_0, \dots, b_{n-1}, 0, \dots, 0)$$
$$p(x) \mapsto (a_0, \dots, a_{n-1})$$
$$q(x) \mapsto (b_0, \dots, b_{n-1})$$

$O(n \cdot \log_2(n))$

$$\textbf{FFT}(\bar{a}) = [y_0, \dots, y_{2n-1}]$$
$$\textbf{FFT}(\bar{b}) = [z_0, \dots, z_{2n-1}]$$
$$p(x) \mapsto \{(\omega_{2n}^0, y_0), \dots, (\omega_{2n}^{2n-1}, y_{2n-1})\}$$
$$q(x) \mapsto \{(\omega_{2n}^0, z_0), \dots, (\omega_{2n}^{2n-1}, z_{2n-1})\}$$

$O(n)$

$$\bar{p} = (y_0 \cdot z_0, y_{2n-1} \cdot z_{2n-1}, \dots, y_1 \cdot z_1)$$
$$p(x) \cdot q(x) \mapsto \{(\omega_{2n}^0, y_0 \cdot z_0), \dots$$
$$\dots, (\omega_{2n}^{2n-1}, y_{2n-1} \cdot z_{2n-1})\}$$

$O(n \cdot \log_2(n))$

$$\frac{1}{2n} \cdot \textbf{FFT}(\bar{p}) = [c_0, \dots, c_{2n-1}]$$
$$p(x) \cdot q(x) \mapsto (c_0, \dots, c_{2n-2})$$