Unfolding of FFT recursion

\[ P_{0,1,\ldots,7}(w_0) = P_{0,2,4,6}(w_0^2) + w_0 P_{1,3,5,7}(w_0^4) \]

\[ P_{0,1,\ldots,7}(w_4) = P_{0,2,4,6}(w_0^2) - w_0 P_{1,3,5,7}(w_0^4) \]

\[ P_{0,2,4,6}(w_0^2) = P_{0,4}(w_0^4) + w_0^2 P_{2,4,6}(w_0^4) \]

\[ P_{0,2,4,6}(w_0^2) = P_{0,4}(w_0^4) - w_0^2 P_{2,4,6}(w_0^4) \]

\[ P_{0,4}(w_0^4) = P_0(w_0^8) + w_0^4 P_4(w_0^8) \]

\[ P_{0,4}(w_0^4) = P_0(w_0^8) - w_0^4 P_4(w_0^8) \]

\[ a_0 \quad a_4 \]
\[ P_A(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} \]
\[ P_B(x) = b_0 + b_1 x + b_2 x^2 + \cdots + b_{n-1} x^{n-1} \]

\[ P_A(x) \times P_B(x) = P_{AB}(x) \]

\[ P_{AB}(x) = a_0 b_0 + (a_0 b_1 + a_1 b_0) x + \]
\[ (a_0 b_2 + a_1 b_1 + a_2 b_0) x^2 + \]
\[ (a_0 b_i + a_i b_{i-1} + \cdots + a_i b_0) x^i \]

This straightforward calculation will take \( O(n^2) \) multiplcations and additions.
\[ P_A(x) \]
\[ \text{deg } n-1 \]

\[ P_B(x) \]
\[ \text{deg } n-1 \]

\[ P_{AB}(x) \]
\[ \text{deg } 2n-2 \]

\[ (x_0, P_A(x_0)) \]

\[ (x_1, P_A(x_1)) \]

\[ \vdots \]

\[ (x_0, P_B(x_0)) \]

\[ (x_1, P_B(x_1)) \]

\[ \vdots \]

\[ (x_0, P_A(x_0), P_B(x_0)) \]

\[ (x_1, P_A(x_1), P_B(x_1)) \]

\[ \vdots \]

\[ O(n \log n) \] time using FFT
\[
\begin{bmatrix}
1 & 1 & 1 & \cdots & 1 \\
w_0 & w_0^2 & w_0^3 & \cdots & w_0^n \\
w_1 & w_1^2 & w_1^3 & \cdots & w_1^n \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
w_i & w_i^2 & w_i^3 & \cdots & w_i^n
\end{bmatrix}
\begin{bmatrix}
a_0 \\
a_1 \\
a_2 \\
\vdots \\
a_{n-1}
\end{bmatrix} =
\begin{bmatrix}
P(1) \\
P(w_1) \\
P(w_2) \\
\vdots \\
P(w_{n-1})
\end{bmatrix}
\]

Polynomial evaluation

\[A = \begin{bmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & w_0 & w_0^2 & \cdots & w_0^n \\
1 & w_1 & w_1^2 & \cdots & w_1^n \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & w_i & w_i^2 & \cdots & w_i^n
\end{bmatrix}, \quad A^{-1} A = I\]

Using

\[1 + w + w^2 + w^{n-1} = 0\]

are also \( n \) roots of \( 2 \) unity
Interpolation also reduces it to an FFT computation, i.e. in \( O(n \log n) \)-time.

Cooley-Tukey algorithm

Application to Schönhage-Strassen multiplication algorithm

Two integers of \( n \) bits can be multiplied in \( O(n \log n \log \log n) \) bit operations.
Given a string $Y$ over some alphabet $\Sigma$, say $\Sigma = \{0, 1\}$ and a pattern $X$, find the occurrences (of any) of $X$ in $Y$.

$|Y| = m$  $|X| = n$  $m \geq n$

$Y = 1101010101010101\ldots$

$X = 1101$

$Y(1) = 1101$

$Y(2) = 1010$

$Y(i)$ denote the substring $Y_i Y_{i+1} \ldots Y_{i+n-1}$

For all $1 \leq i \leq m-n+1$

Check if $Y(i) = X$ (by bit by bit check)

Time in $O((m-n)n) \approx O(mn)$

Ideal bound $O(mn)$

$Y(i)$ and $Y(i+1)$ differ by 2 bits.
\# \gamma(i+1) = \# \gamma(i) \times 2 - 2^n + Y_{i+n}

the number denoted by the story

We are still dealing with \( n \) lit numbers where \( n \) can be large.