CSL 356 Lecture 33 Oct 18

FFT → contd

Unfolding of FFT recursion

$$P_{0,1..7}(\omega_0) = P_{0,2,4,6}(\omega_0^2) + \omega_0 P_{1,3,5,7}(\omega_0^2)$$

$$P_{0,1,..7}(\omega_4) = P_{0,2,4,6}(\omega_0^2) - \omega_0 P_{1,3,5,7}(\omega_0^2)$$

$$P_{0,2,4,6}(\omega_0^2) = P_{0,4}(\omega_0^4) + \omega_0^2 P_{2,6}(\omega_0^4)$$

$$P_{0,2,4,6}(\omega_2^2) = P_{0,4}(\omega_0^4) - \omega_0^2 P_{2,6}(\omega_0^4)$$

$$P_{0,4}(\omega_0^4) = P_0(\omega_0^8) + \omega_0^4 P_4(\omega_0^8)$$

$$P_{0,4}(\omega_1^4) = \underbrace{P_0}_{a_0}(\omega_0^8) - \omega_0^4 \underbrace{P_4}_{a_4}(\omega_0^8)$$

$$P_A(x) = a_0 + a_1 x + a_2 x^2 + \cdots \quad a_{n-1} x^{n-1}$$

$$P_B(x) = b_0 + b_1 x + b_2 x^2 \quad \cdots \quad b_{n-1} x^{n-1}$$

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

$$P_{AB}(x) = a_0 b_0 + (a_0 b_1 + a_1 b_0) x +$$

$$(a_0 b_2 + a_1 b_1 + a_2 b_0) x^2 +$$

$$\vdots$$

$$(a_0 b_i + a_1 b_{i-1} + \cdots \cdot a_i b_0) x^i$$

$$\vdots$$

This straightforward calculation will take $O(n^2)$ mult & additions

$P_A(x)$ $\left.\begin{array}{l}(x_0, \; P_A(x_0)) \\ (x_1, \; P_A(x_1)) \\ \vdots \\ (x_0, \; P_B(x_0)) \\ (x_1, \; P_B(x_1))\end{array}\right\}$ $O(n\log n)$ time using FFT

deg $n-1$

$P_B(x)$

deg $n-1$

$\vdots$

$P_{AB}(x)$ $\left.\begin{array}{l}(x_0, \; P_A(x_0) \cdot P_B(x_0)) \\ (x_1, \; P_A(x_1) \cdot P_B(x_1)) \\ \vdots\end{array}\right\}$ $O(n)$

deg $2n-2$

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_0 & \omega_0^2 & \omega_0^3 & \cdots & \omega_0^{n-1} \\ 1 & \omega_1 & \omega_1^2 & \cdots & & \omega_1^{n-1} \\ 1 & \omega_i & \omega_i^2 & \cdots & & \omega_i^{n-1} \\ \vdots & & & & & \end{bmatrix}}_{= A} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{bmatrix}}_{\bar{a}} = \underbrace{\begin{bmatrix} P(1) \\ P(\omega_1) \\ P(\omega_2) \\ \vdots \\ P(\omega_{n-1}) \end{bmatrix}}_{\bar{P}}$$

Polynomial evaluation

$$\omega_i = \omega^i$$

Interpolation $\quad \bar{a} = A^{-1} \bar{P}$

$$A^{-1} = \frac{1}{n} \begin{bmatrix} 1 & 1 & 1 & \cdots & & 1 \\ 1 & \frac{1}{\omega_0} & \frac{1}{\omega_0^2} & \frac{1}{\omega_0^3} & \cdots & \frac{1}{\omega_0^{n-1}} \\ \vdots & & & & & \\ 1 & \frac{1}{\omega_i} & \frac{1}{\omega_i^2} & \cdots & & \frac{1}{\omega_i^{n-1}} \\ \vdots & & & & & \end{bmatrix}$$

$$A^{-1} A = I$$

using
$1 + \omega + \omega^2 + \cdots \omega^{n-1} = 0$

$1, \frac{1}{\omega_i}, \frac{1}{\omega_i^2} \cdots$ are also n^{th} roots of unity

Interpolation also reduces to an FFT computation, i.e. in $O(n \log n)$-time

Cooley - Tukey algorithm

Application to Schonage - 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 Σ, say $\Sigma = \{0, 1\}$

and a pattern X, find the occurrences (if any) of X in Y

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

$$
\begin{array}{lccccccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & - & & - & - & - & . & m \\
Y = & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & \ldots \\
X = & 1 & 0 & 0 & 1 \\
 & & 1 & 0 & 0 & 1
\end{array}
$$

$Y(1) = 1\,1\,0\,1$

$Y(2) = \;\;\;1\,0\,1\,0$

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

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

check if $Y(i) = X$ /* bit by bit check */

Time is $O((m-n)n) \sim O(mn)$

Ideal bound $O(m+n)$

$\boxed{Y(i) \text{ and } Y(i+1) \text{ differ by 2 bits}}$

$$\#\, Y(i+1) = \#\, Y(i) \times 2 - 2^n + Y_{i+n}$$

the number denoted by the string

We are still dealing with n bit numbers where n can be large