CSE525 Lec5: FFT

Debajyoti Bera (M21)

$P(x) = 5x^5 + 7x^4 - 3x^2 + 3$

$p(100) = ?$
$p(200) = \ldots$
$p(73) = \ldots$

Polynomial evaluation

Given a degree (n-1) polynomial

original

$\boxed{P_n(x)} = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + \ldots + a_{n-1} x^{n-1}$

... and a number y, compute P(y)

... and a list of numbers $y_1, y_2, \ldots y_k$, compute $P(y_1), P(y_2), \ldots P(y_k)$

Suppose we knew how to solve a smaller polynomial evaluation problem.

$P(y)$ for $y < x$ ☹

$P'(y)$ for P' of deg $< n-1$

$p'(x) = 5x^4 + 7x^3 - 3x$ at $100 = q$

$P(100) = p'(100) \times 100 + 3$

$T(n) = T(n-1) + \underbrace{O(1)}$
$= O(n)$

Divide-&-Conquer for single evaluation

$P_n(x) = a_0 + a_1\, x + a_2\, x^2 + a_3\, x^3 + a_4\, x^4 + \ldots + a_{n-1}\, x^{n-1}$

Eval(P(), x, n) : input a degree-(n-1) polynomial P() and a number x. Computes P(x)

Recursive algorithm for Eval()? Complexity of approach ?

Divide-&-Conquer for single evaluation

$P(x) = x^{100} + x^{99} + \cdots + 1$

$Odd(x) = x^{99} + x^{97} + \cdots$

$Odd(x) = 5x^3$

$P(100) = Odd(100) \times 100^2$

Assume we know how to evaluate a smaller deg. poly. on any number.

$P_n(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + \ldots + a_{n-1} x^{n-1}$

Eval(P(), x, n) : input a degree-(n-1) polynomial P() and a number x. Computes P(x)

$P(100) = 5 \times 100^5 + 7 \times 100^4 - 3 \times 100^2 + 3$

Recursive algorithm for Eval()? Complexity of approach ?

$\frac{n-3}{2}$

$Odd(x) = a_1 + a_3 x + a_5 x^2 + \ldots + a_{n-2} x^{(n-3)/2}$

$Even(x) = a_0 + a_2 x + a_4 x^2 + \ldots + a_{n-1} x^{(n-1)/2}$

$Odd(x^2) = a_1 + a_3 x^2 + a_5 x^4 + \cdots + a_{n-2} x^{n-3}$

Use Eval to evaluate $Odd(y^2)$ and $Even(y^2)$:

$P_n(y) = Even(y^2) + y*Odd(y^2)$

$Odd(x^2) \times x = a_1 x + a_3 x^3 + a_5 x^5 + \cdots + a_{n-2} x^{n-2}$

$Even(x^2) = a_0 + a_2 x^2 + \cdots + a_{n-1} x^{(n-1)/2}$

$x * Odd(x^2) + E(x^2) = P(x)$

$P(x) = 5x^5 + 7x^4 - 3x^2 + 3x^0$

$5x^5$ $\quad 7x^4 - 3x^2 + 3$

$Odd(x) = 5x^2$ $\quad Even(x) = 7x^2 - 3x + 3$

$P(100) = Odd(100^2) * 100 + Even(100^2)$

$T(n) = 2T(n/2) + O(1)$

$= O(n)$

Divide-&-Conquer for single evaluation

$High(x) = 5x^5 + 7x^4$

↳ degree is not reducing

$P_n(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + \dots + a_{n-1} x^{n-1}$

n ↳ # coefficients

Eval(P(), x, n) : input a degree-(n-1) polynomial P() and a number x. Computes P(x)

Recursive algorithm for Eval()? Complexity of approach ?

$n=6$

$P(x) = \boxed{5x^5 + 7x^4} - 3x^2 + 3$

$x^2 * (5x^3 + 7x^2)$

$\left.\begin{array}{l} Low(x) = -3x^2 + 3 \\ High(x) = 5x^3 + 7x^2 \end{array}\right\}$ deg $\le \frac{n}{2} - 1$

$P(x) = Low(x) + High(x) * x^2$

$T(n) = 2T(n/2) + O(1)$

$Low(x) = a_0 + a_1 x + \dots + a_{n/2-1}\, x^{n/2-1}$
$High(x) = a_{n/2} + a_{1+n/2}\, x + \dots + a_{n-1} x^{n/2-1}$

$Single(x) = x^{n/2}$

Use Eval to evaluate Low(y) & High(y) & Single(y).
$P_n(y) = Low(y) + Single(y)*High(y)$

$\omega = e^{2\pi i/3}$ $\omega^3 = 1$, $(\omega^2)^3 = 1$ There a K k-th roots of 1

$\omega = e^{2\pi i/k}$ $\omega^0, \omega^1, \dots, \omega^{k-1}$

$= 1$

$\omega^k = e^{2\pi i/k \cdot k} = e^{2\pi i} = 1$

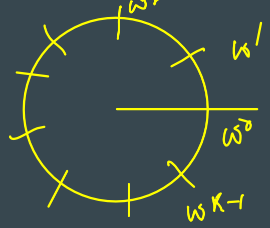
Discrete Fourier Transform

Input: coefficients of a degree-15 polynomial $A_{16}(x)$: $[a_0 \ a_1 \ \dots \ a_{15}]$

DFT of A of order k : $\mathrm{DFT}_8(A_{16}) = [A_{16}(1) \ A_{16}(w) \ A_{16}(w^2) \ \dots \ A_{16}(w^7)]$

ω_8^0 ω_8^1 ω_8^2 ω_8^7

where, w : 8-th root of unity



DFT of A : DFT of order deg(A)+1

$A(x) = 3x^2 - 7x + 1$

Q: How to obtain $\mathrm{DFT}_8(A_{16})$ from A_{16} ?

Evaluation

$\mathrm{DFT}_3(A) = [3 - 7 + 1 = -3, \ 3\omega_3^2 - 7\omega_3 + 1, \ 3(\omega_3^2)^2 - 7\omega_3^2 + 1]$

DFT is a <u>fingerprint</u> of polynomial A(x)

Q: How to obtain A_{16} from $\mathrm{DFT}_{\cancel{8}\,16}(A_{16})$? Interpolation

$w_8^{10} = e^{2\pi i \cdot 10/8} = \underbrace{e^{2\pi i}}_{1} \cdot e^{2\pi i \cdot 2/8} = (e^{2\pi i/8})^2 = w_8^2$

$w_8^2 = e^{2\pi i \cdot 2/8} = e^{2\pi i \cdot 1/4}$

$w_8^4 = e^{2\pi i \cdot 2/4}$

$w_8^6 = e^{2\pi i \cdot 3/4}$

$w_8^8 = 1$

Eval. $DFT_8(A_{16}) = < A_{16}(w_8^0), A_{16}(w_8^1), A_{16}(w_8^2), \ldots, A_{16}(w_8^7) >$

$= 1$

$w_8^0 = w_4^0$

want

A(1)	$\boxed{\text{Odd}(1)}$	Even(1)	$1*O(1) + E(1)$
$A(w_8)$	$Odd(w^2)$	$Even(w^2)$	$w_8*O(w_8^2) + E(w_8^2)$
$A(w^2)$	$Odd(w^4)$	$Even(w^4)$	$w_8^2*O(w_8^4) + E(w_8^4)$
$A(w^3)$	$Odd(w^6)$	$Even(w^6)$	$w_8^3*O(w_8^6) + E(w_8^6)$
$A(w^4)$	$Odd(w_8^8)$ $= Odd(1)$	$Even(w^8)$	...
$A(w^5)$	$Odd(w_8^{10})$	$Even(w_8^{10})$	
$A(w^6)$	$Odd(w_8^{12})$	...	...
$A(w^7)$	$Odd(w_8^{14})$	...	...

O(k) Steps

k: DFT order

$A_n(y) = Even_{n/2}(y^2) + y*Odd_{n/2}(y^2)$

$w_8^{8+k} = w^k \qquad w_8^{2k} = w_4^k$

Q: Express $DFT_8(A_{16})$ in terms of $DFT_4(???)$ problems.

Overall problem size is halved !

Compute $Odd(w_8^0), Odd(w_8^2)$
$Odd(w_8^4), Odd(w_8^6)$
$Even(\ldots)$

$DFT_4(Odds)$
Compute $Odd(w_4^0), Odd(w_4), Odd(w_4^2),$
$Odd(w_4^3), Even(\ldots)$

$DFT_8(A)$	$DFT_4(???)$	$DFT_4(???)$

$e^{\pi i}$ e^0 $\}$ 2 2-roots of 1

Eval. $DFT_8(A_{16}) = \langle A_{16}(w_8^{\,0}), A_{16}(w_8^{\,1}), A_{16}(w_8^{\,2}), \ldots, A_{16}(w_8^{\,7}) \rangle$

A(1)	Odd(1)	Even(1)	1*O(1) + E(1)
A(w)	$Odd(w^2)$	$Even(w^2)$	$w_8 * O(w_8^{\,2}) + E(w_8^{\,2})$
$A(w^2)$	$Odd(w^4)$	$Even(w^4)$	$w_8^{\,2} * O(w_8^{\,4}) + E(w_8^{\,4})$
$A(w^3)$	$Odd(w^6)$	$Even(w^6)$	$w_8^{\,3} * O(w_8^{\,6}) + E(w_8^{\,6})$
…	…	…	…
$A(w^7)$	…	…	…

$DFT_8(A)$ $DFT_4(???)$ $DFT_4(???)$

$$A_n(y) = Even_{n/2}(y^2) + y*Odd_{n/2}(y^2)$$

$w_8^{\,8+k} = w^k$ $\quad w_8^{\,2k} = w_4^{\,k}$

Complexity analysis:

$T(n,k)$: $DFT_k(A_n)$

$= 2T(n/2, k/2) + O(k)$

Compute $DFT_n(A_n)$

$T(n) = 2T(n/2) + O(n)$

$= O(n \lg n)$

FFT algorithm for DFT

Algorithm $\mathrm{FFT}_n(\langle a_0, \ldots, a_{n-1} \rangle)$ DFT_n (n-coeff. poly)

if $n = 1$ **then return** $\langle a_0 \rangle$
else
    $\omega_n \leftarrow e^{2\pi i/n}$
    $\omega \leftarrow 1$
    $\langle y_0^{\text{even}}, \ldots, y_{n/2-1}^{\text{even}} \rangle \leftarrow \mathrm{FFT}_{n/2}(\langle a_0, a_2, \ldots, a_{n-2} \rangle)$
    $\langle y_0^{\text{odd}}, \ldots, y_{n/2-1}^{\text{odd}} \rangle \leftarrow \mathrm{FFT}_{n/2}(\langle a_1, a_3, \ldots, a_{n-1} \rangle)$
    for $k \leftarrow 0$ **to** $n/2 - 1$ **do**
        $y_k \leftarrow y_k^{\text{even}} + \omega y_k^{\text{odd}}$
        $y_{k+n/2} \leftarrow y_k^{\text{even}} - \omega y_k^{\text{odd}}$
        $\omega \leftarrow \omega \omega_n$
    return $\langle y_0, \ldots, y_{n-1} \rangle$

Complexity ?

Linear Algebraic form of Discrete Fourier Transform

DFT: Map $[x_0 \ldots x_{N-1}]$ to $[X_0 \ldots X_{N-1}]$

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn}$$

$$= \sum_{n=0}^{N-1} x_n \cdot [\cos(2\pi kn/N) - i \cdot \sin(2\pi kn/N)],$$

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2-i \\ -i \\ -1+2i \end{pmatrix}$$

$$\longrightarrow \qquad \mathbf{X} = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} 2 \\ -2-2i \\ -2i \\ 4+4i \end{pmatrix}$$

wikipedia.com

$$DFT_n \begin{bmatrix} B(\omega^0) \\ B(\omega^1) \\ \vdots \\ B(\omega^{n-1}) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} \xleftarrow[O(n \lg n)]{\text{interpolation}} \xrightarrow[O(n \lg n)]{\text{mult. } V_n} \begin{bmatrix} A(\omega^0) \\ A(\omega^1) \\ \vdots \\ A(\omega^{n-1}) \end{bmatrix}$$

$$V_n \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}}_{\text{coeff of } A(x)} = \begin{bmatrix} a_0 + a_1 + \cdots + a_{n-1} = A(\omega_n^0) \\ a_0 + a_1\omega_n + a_2\omega_n^2 + \cdots = A(\omega_n^1) \\ a_0 + a_1\omega_n^2 + a_2\omega_n^4 + \cdots = A(\omega_n^2) \\ \vdots \\ a_0 + a_1\omega_n^{n-1} + a_2(\omega_n^{n-1})^2 + \cdots = A(\omega_n^{n-1}) \end{bmatrix}$$

DFT $(4 + 5x - x^2)$ = ?

DFT $^{-1}$ ([2, -2-2i, -2i, 4+4i]) = ?

$$V_n \vec{A} = \overrightarrow{DFT(A)}$$

$$\vec{A} = V_n^{-1} \overrightarrow{DFT(A)}$$

Inverse DFT ?

- Is DFT invertible ?
- If yes, how to invert ?

$$V_n \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix} = \ ?$$

$$A(x): \quad V_n = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)(n-1)} \end{pmatrix} \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

$B(x) = b_0 + b_1 x + b_2 x^2 + \cdots$

$$V_n^{-1} = \frac{1}{n} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n^{-1} & \omega_n^{-2} & \cdots & \omega_n^{-(n-1)} \\ 1 & \omega_n^{-2} & \omega_n^{-4} & \cdots & \omega_n^{-2(n-1)} \\ 1 & \omega_n^{-3} & \omega_n^{-6} & \cdots & \omega_n^{-3(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{-(n-1)} & \omega_n^{-2(n-1)} & \cdots & \omega_n^{-(n-1)(n-1)} \end{pmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \end{bmatrix}$$

- $V_n V_n^{-1} = \ ?$
- Take row j (j=0,1, …)

  $w_n^{-jk} = \ ?$

$A(\omega^1) = a_0 + a_1\omega + a_2\omega^2 + \cdots$

$V_n^{-1} \times \begin{bmatrix} A(\omega_0) \\ \vdots \end{bmatrix} = \begin{bmatrix} a_0 \\ \vdots \end{bmatrix} \rightarrow a_1 = b_0 + b_1 \omega^{-1} + b_2 \omega^{-2} = B(\omega_n^{-1})$