CS F364: Design & Analysis of Algorithm

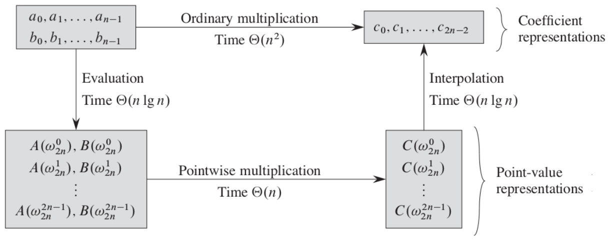
05 Discrete and Fast Fourier Transform

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Jan 25, 2021 ONLINE (Campus @ BITS-Pilani Jan-May 2021)

http://ktiwari.in/algo

Polynomial Multiplication



n^{th} root of unity

Roots are $\omega_n^0 = 1, \omega_n^1, \omega_n^2, \ldots, \omega_n^{n/2} = -1, \ldots, \omega_n^{n-1}$

$$\omega_n = e^{2\pi i/n} \qquad e^{i\theta} = \cos(\theta) + i.\sin(\theta)$$

- $\omega_{dn}^{dk} = \omega_n^k$
- $\omega_n^{n/2} = \omega_2 = -1$
- Halving lemma: $(\omega_n^{k+n/2})^2 = (\omega_n^k)^2$ or $\omega_n^{k+n/2} = -\omega_n^k$
- Summation lemma:

$$\sum_{j=0}^{n-1} (\omega_n^k)^j = \frac{(\omega_n^k)^n - 1}{\omega_n^k - 1} = \frac{(\omega_n^n)^k - 1}{\omega_n^k - 1} = \frac{(1)^n - 1}{\omega_n^k - 1} = \frac{1-1}{\omega_n^k - 1} = 0$$

DFT - Discrete Fourier Transform

Convert coefficient form of a polynomial $a = (a_0, a_1, a_2, \ldots, a_{n-1})$ to point value form $y = (y_0, y_1, y_2, \ldots, y_{n-1})$ where $y_k = A(\omega_n^k)$

$$y_k = \sum_{j=0}^{n-1} a_j \times (\omega_n^k)^j$$

- $y = DFT_n(a)$ we say y is DFT of a

FFT - Fast Fourier Transform

$$A^{[0]}(x) = a_0 + a_2 x + a_4 x^2 + \ldots + a_{n-2} x^{n/2+1}$$
$$A^{[1]}(x) = a_1 + a_3 x + a_5 x^2 + \ldots + a_{n-1} x^{n/2+1}$$

$$\boxed{A(x) = A^{[0]}(x^2) + x.A^{[1]}(x^2)}$$

Recursive FFT

Algorithm 1: Rec-FFT (a)

```
n ← length(a)
If (n = 1) return a
a^[0] = (a_0, a_2, a_4, a_{n-2})     y^[0] = Rec-FFT(a^[0])
a^[1] = (a_1, a_3, a_5, a_{n-1})     y^[1] = Rec-FFT(a^[1])
ω_n ← e^{2πi/n},   ω ← 1
for k ← 0 to n/2 − 1 do
    y_k = y_k^[0] + ω.y_k^[1]
    y_{k+(n/2)} = y_k^[0] − ω.y_k^[1]
    ω ← ω.ω_k
return y
```


Recursive FFT

Algorithm 2: Rec-FFT (a)

```
n ← length(a)
If (n = 1) return a
a^[0] = (a_0, a_2, a_4, a_{n-2})     y^[0] = Rec-FFT(a^[0])
a^[1] = (a_1, a_3, a_5, a_{n-1})     y^[1] = Rec-FFT(a^[1])
ω_n ← e^{2πi/n},   ω ← 1
for k ← 0 to n/2 − 1 do
    y_k = y_k^[0] + ω.y_k^[1]
    y_{k+(n/2)} = y_k^[0] − ω.y_k^[1]
    ω ← ω.ω_k
return y
```

Time complexity $T(n) = 2T(n/2) + c.n$

$O(n. \log n)$

FFT

$$y_k^{[0]} = A^{[0]}(\omega_{n/2}^k) ,$$
$$y_k^{[1]} = A^{[1]}(\omega_{n/2}^k) ,$$

or, since $\omega_{n/2}^k = \omega_n^{2k}$ by the cancellation lemma,

$$y_k^{[0]} = A^{[0]}(\omega_n^{2k}) ,$$
$$y_k^{[1]} = A^{[1]}(\omega_n^{2k}) .$$

$$\begin{aligned} y_k &= y_k^{[0]} + \omega_n^k y_k^{[1]} \\ &= A^{[0]}(\omega_n^{2k}) + \omega_n^k A^{[1]}(\omega_n^{2k}) \\ &= A(\omega_n^k) && \text{(by equation (30.9))} . \end{aligned}$$

For $y_{n/2}, y_{n/2+1}, \ldots, y_{n-1}$, letting $k = 0, 1, \ldots, n/2 - 1$, line 12 yields

$$\begin{aligned} y_{k+(n/2)} &= y_k^{[0]} - \omega_n^k y_k^{[1]} \\ &= y_k^{[0]} + \omega_n^{k+(n/2)} y_k^{[1]} && (\text{since } \omega_n^{k+(n/2)} = -\omega_n^k) \\ &= A^{[0]}(\omega_n^{2k}) + \omega_n^{k+(n/2)} A^{[1]}(\omega_n^{2k}) \\ &= A^{[0]}(\omega_n^{2k+n}) + \omega_n^{k+(n/2)} A^{[1]}(\omega_n^{2k+n}) && (\text{since } \omega_n^{2k+n} = \omega_n^{2k}) \\ &= A(\omega_n^{k+(n/2)}) && \text{(by equation (30.9))} . \end{aligned}$$

Inverse DFT

Convert $y = (y_0, y_1, y_2, ..., y_{n-1})$ to polynomial $a = (a_0, a_1, a_2, ..., a_{n-1})$

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \omega_n^3 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \omega_n^6 & \cdots & \omega_n^{2(n-1)} \\ 1 & \omega_n^3 & \omega_n^6 & \omega_n^9 & \cdots & \omega_n^{3(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \omega_n^{3(n-1)} & \cdots & \omega_n^{(n-1)(n-1)} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

- $y = V_n a$
- $a = DFT_n^{-1}(y)$
- $a = V_n^{-1} y$

- (k, j) entry of V is ω_n^{kj} where $k, j \in \{0, 1, .., n-1\}$

Inverse: consider $V_n^{-1} V_n = I$

$$V_n^{-1}(i, j) = \omega_n^{-ij}/n$$

- Apply same method to find multiplication in $O(n \log n)$ time

Thank You!

Thank you very much for your attention! (Reference[1])

Queries ?

[1] [1] Book - *Introduction to Algorithm*, By THOMAS H. CORMEN, CHARLES E. LEISERSON, RONALD L. RIVEST, CLIFFORD STEIN