Divide and Conquer 5: FFT

FFT: High-Level

1. Goal: Evaluate polynomial A(x) of degree <= n-1 at n points
    - N points: n^{th} roots of unity
        - $n = 2^k$
    - Define $A_{even}(y)$ and $A_{odd}(y)$ of degree <= n/2-1
    - Recursively evaluate A_{even} and A_{odd} at $(n^{th}\text{ roots})^2$
    - Then, O(n) time to get A(x) at n^{th} roots
        - $A(x) = A_{even}(x^2) + xA_{odd}(x^2)$
    - T(n) = 2T(n/2) + O(n) = O(nlogn)

FFT: Pseudocode

1. FFT(a,w):
    - Input: coefficients $a = (a_0, a_1, \ldots, a_{n-1})$ for polynomial A(x) where n is a power of 2
        - w is a n^{th} root of unity
        - Use $w = w_n = (1, 2pi/n) = e^{(2ipi/n)}$
    - Output: $A(w^0), A(w), A(w^2), \ldots, A(w^{n-1})$

FFT: Core

```
FFT(a,w):
    if n == 1:
        return A(1)
    Let Aeven = (a0,a2,a4,...an-2)
    Let Aodd = (a1, a3, ..., an-1)
    Call FFT(Aeven, w^2): get Aeven(w0), Aeven(w2), ..., Aeven(w^(n-2))
    Call FFT(Aodd, w^2): get Aodd(w0), ..., Aodd(w^(n-2))
    # if w == wn then (wn^j)^2 = (wn/2)^j
    for j = 0 to n/2-1:
        A(wj) = Aeven(w^2j) + w^jAodd(w^2j)
        A(w^(n/2+j)) = A(-w^j) = Aeven(w^2j) - w^j * Aodd(w^2j)
    return [A(w0), A(w1) ..., A(wn-1)]
```

$\underline{\text{FFT}(a, \omega):}$

if $n = 1$, return$(A(1))$

Let $a_{even} = (a_0, a_2, a_4, \ldots, a_{n-2})$ & $a_{odd} = (a_1, a_3, \ldots, a_{n-1})$

Call FFT(a_{even}, ω^2): get $A_{even}(\omega^0), A_{even}(\omega^2), \ldots, A_{even}(\omega^{n-2})$

Call FFT(a_{odd}, ω^2): get $A_{odd}(\omega^0), \ldots, A_{odd}(\omega^{n-2})$

For $j = 0 \to \frac{n}{2} - 1$: $A(\omega^j) = A_{even}(\omega^{2j}) + \omega^j A_{odd}(\omega^{2j})$

$A(\omega^{\frac{n}{2}+j}) = A(-\omega^j) = A_{even}(\omega^{2j}) - \omega^j A_{odd}(\omega^{2j})$

Return$(A(\omega^0), A(\omega^1), \ldots, A(\omega^{n-1}))$

FFT Pseudocode

FFT: Concise

1. Part of the appeal of the FFT algorithm is how concise it is

$\underline{\text{FFT}(a, \omega):}$

if $n = 1$, return(a_0)

Let $a_{even} = (a_0, a_2, \ldots, a_{n-2})$ & $a_{odd} = (a_1, a_3, \ldots, a_{n-1})$

$(s_0, s_1, \ldots, s_{\frac{n}{2}-1}) = \text{FFT}(a_{even}, \omega^2)$

$(t_0, t_1, \ldots, t_{\frac{n}{2}-1}) = \text{FFT}(a_{odd}, \omega^2)$

For $j = 0 \to \frac{n}{2} - 1$: $r_j = s_j + \omega^j t_j$

$r_{\frac{n}{2}+j} = s_j - \omega^j t_j$

Return$(r_0, r_1, \ldots, r_{n-1})$

FFT Pseudocode Concise

FFT: Running Time

1. T(n) = 2T(n/2) + O(n) = O(nlogn)

Polynomial Multiplication using FFT

1. Input:
   - Polynomials A(x) and B(x)
     - $A(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{n-1}x^{n-1}$
     - $B(x) = b_0 + b_1x + b_2x^2 + \ldots + b_{n-1}x^{n-1}$
2. Output:
   - Want $C(x) = A(x)B(x)$
     - $C(x) = c_0 + c_1x + c_2x^2 + \ldots + c_{2n-2}x^{2n-2}$
     - $c_k = a_0b_k + a_1b_{k-1} + \ldots + a_kb_0$
3. Procedure:
   - $(r_0, r_1, \ldots, r_{2n-1}) = \text{FFT}(a, w_{2n})$
   - $(s_0, s_1, \ldots, s_{2n-1}) = \text{FFT}(b, w_{2n})$
   - for j=0 -> 2n-1: t[j] = r[j] * s[j]
   - Have C(x) at $2n^{th}$ roots of unity: Run inverse FFT to get coefficients

Linear Algebra View

1. For point x_j: $A(x_j) = a_0 + a_1x_j + \ldots + a_{n-1}x_j^{n-1}$
   - $A(x_j) = (1, x_j, \ldots, x_j^{n-1}) * (a_0, a_1, \ldots, a_{n-1})$

For point x_j: $A(x_j) = a_0 + a_1 x_j + a_2 x_j^2 + \cdots + a_{n-1} x_j^{n-1}$

$$= (1, x_j, x_j^2, \ldots, x_j^{n-1}) \cdot (a_0, a_1, \ldots, a_{n-1})$$

For points $x_0, x_1, \ldots, x_{n-1}$:

$$\begin{bmatrix} A(x_0) \\ A(x_1) \\ \vdots \\ A(x_{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ & & \vdots & & \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

Linear Algebra View

LA View of FFT

$$\begin{bmatrix} A(x_0) \\ A(x_1) \\ \vdots \\ A(x_{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ & & \vdots & & \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

Let $x_j = \omega_n^j$

$$\begin{bmatrix} A(1) \\ A(\omega_n) \\ \vdots \\ A(\omega_n^{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ & & \vdots & & \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

Linear Algebra View of FFT

LA for Inverse FFT

$$\begin{bmatrix} A(1) \\ A(\omega_n) \\ \vdots \\ A(\omega_n^{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ & & \vdots & & \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

$$\underset{A}{\shortparallel} \qquad\qquad \underset{M_n(\omega_n)}{\shortparallel} \qquad\qquad \underset{a}{\shortparallel}$$

$$A = M_n(\omega_n)\,a = FFT(a, \omega_n)$$

$$M_n(\omega_n)^{-1} A = a$$

Inverse FFT

1. Lemma: $M_n(w_n)^{-1} = 1/n * M_n(w_n^{-1})$
   - What is w_n^{-1}?
     - $w_n * w_n^{-1} = 1$
     - $w_n * w_n^{n-1} = w_n^{n} = w_n^{0} = 1$
   - $M_n(w_n)^{-1} = 1/n * M_n(w_n^{n-1})$

Inverse FFT via FFT

1. Lemma: $M_n(w_n)^{-1} = 1/n * M_n(w_n^{-1}) = M_n(w_n)^{-1} = 1/n * M_n(w_n^{n-1})$
   - $na = M_n(w_n^{n-1})A = FFT(A, w_n^{n-1})$
   - $a = 1/n * FFT(A, w_n^{n-1})$

Quiz: Inverses

1. What is (w_n^{2})-1? For what power k is $(w_n)^k * (w_n)^2 = 1$?
   - $k = n - 2$
   - $w_n^{2} * w_n^{n-2} = w_n^{n} = 1$

Quiz: Sum of Roots

1. For even n, $1 + w_n + w_n^2 + \ldots + w_n^{n-1}$?
    - 0
    - $w_n^j = -w_n^{n/2+j}$

Proof of Claim (FFT)

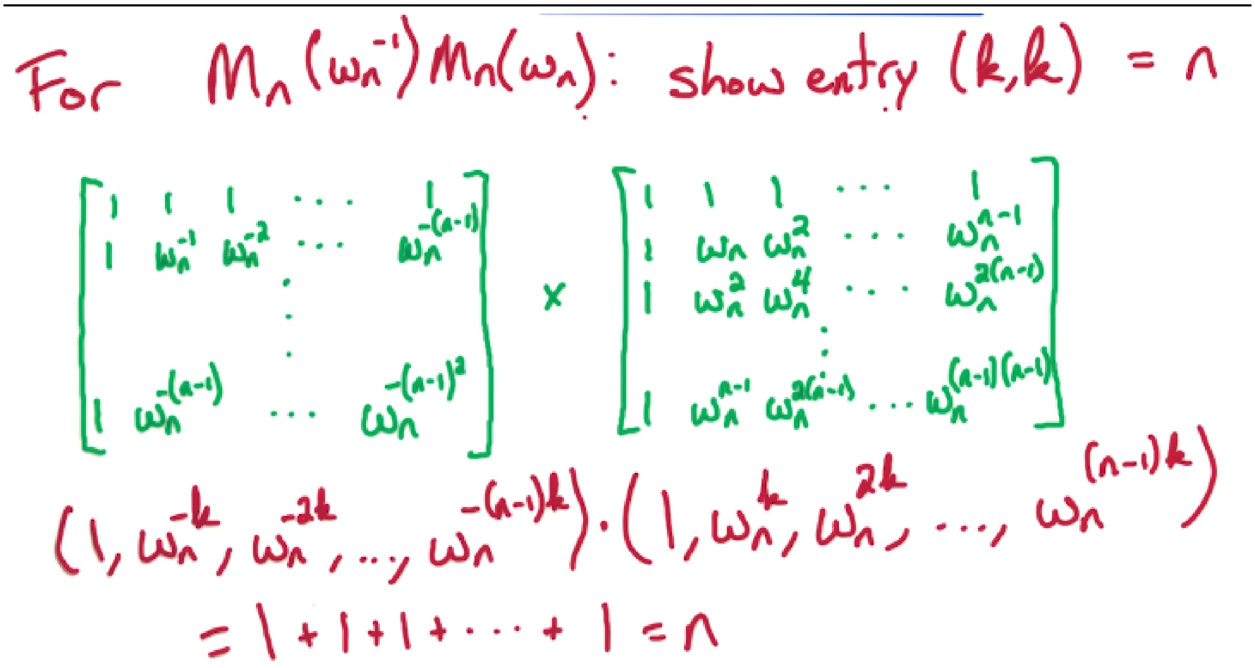
1. Claim: For any n^{th} root of unity w where w != 1:
    - $1 + w + w^2 + \ldots + w^{n-1} = 0$
2. Proof: For any number z $(z-1)(1+z+z^2+\ldots+z^{n-1})$
    - $= (z + z^2 + \ldots + z^n) - (1 + z + \ldots + z^{n-1})$
    - $= z^n - 1$
    - Let z = w:
        - $z^n = 1$
        - Because we assume w != 1, z-1 != 0, so the second term must equal 0

Proof of Lemma

1. Need to show:
    - $M_n(w_n)^{-1} = 1/n * M_n(w_n^{-1})$
    - $1/n * M_n(w_n^{-1})M_n(w_n) = I$
2. For $M_n(w_n^{-1})M_n(w_n)$:
    - Show entries (k,k) are n and for k != j (k,j) are 0

Diagonal Entries

1. For $M_n(w_n^{-1})M_n(w_n)$: Show entry (k,k) = n



Diagonal Entries

Off-Diagonal Entries

1. For $M_n(w_n^{-1})M_n(w_n)$: Show entry $(k,j) = 0$

For $M_n(\omega_n^{-1})M_n(\omega_n)$: show entry $(k,j) = 0$
for $k \neq j$

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n^{-1} & \omega_n^{-2} & \cdots & \omega_n^{-(n-1)} \\ \vdots & & & & \vdots \\ 1 & \omega_n^{-(n-1)} & & \cdots & \omega_n^{-(n-1)^2} \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \cdots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \cdots & \omega_n^{2(n-1)} \\ \vdots & & \vdots & & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \cdots & \omega_n^{(n-1)(n-1)} \end{bmatrix} \qquad \omega = \omega_n^{j-k} \neq 1$$

$$(1, \omega_n^{-k}, \omega_n^{-2k}, \dots, \omega_n^{-(n-1)k}) \cdot (1, \omega_n^{j}, \omega_n^{2j}, \dots, \omega_n^{(n-1)j})$$

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

Off-diagonal Entries

Back to Polynomial Multiplication

1. Input:
   - Polynomials A(x) and B(x)
     - $A(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1}$
     - $B(x) = b_0 + b_1x + b_2x^2 + \dots + b_{n-1}x^{n-1}$
2. Output:
   - Want $C(x) = A(x)B(x)$
     - $C(x) = c_0 + c_1x + c_2x^2 + \dots + c_{2n-2}x^{2n-2}$
     - $c_k = a_0b_k + a_1b_{k-1} + \dots + a_kb_0$
3. Procedure:
   - $(r_0, r_1, \dots, r_{2n-1}) = FFT(a, w_{2n})$
   - $(s_0, s_1, \dots, s_{2n-1}) = FFT(b, w_{2n})$
   - for j=0 -> 2n-1: t[j] = r[j] * s[j]
   - Have C(x) at $2n^{th}$ roots of unity: Run inverse FFT to get coefficients
     - $(c_0, \dots, c_{2n-1}) = 1/2n * FFT(t, w_{2n}^{2n-1})$