COL351: Analysis and Design of Algorithms

Ragesh Jaiswal, CSE, IITD

Divide and Conquer

Problem

Given two polynomials:
$A(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + ... + a_{n-1} \cdot x^{n-1}$, and
$B(x) = b_0 + b_1 \cdot x + b_2 \cdot x^2 + ... + b_{n-1} \cdot x^{n-1}$, design an algorithm to that outputs $A(x) \cdot B(x)$.

Problem

Given two polynomials:
$A(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + ... + a_{n-1} \cdot x^{n-1}$, and
$B(x) = b_0 + b_1 \cdot x + b_2 \cdot x^2 + ... + b_{n-1} \cdot x^{n-1}$, design an algorithm to that outputs $A(x) \cdot B(x)$.

- We have to obtain the polynomial $C(x) = A(x) \cdot B(x)$
- $C(x)$ may be written as:
  $C(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + ... + c_{2n-2} \cdot x^{2n-2}$
- What is c_i in terms of coefficients of A and B?

Problem

Given two polynomials:
$A(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + ... + a_{n-1} \cdot x^{n-1}$, and
$B(x) = b_0 + b_1 \cdot x + b_2 \cdot x^2 + ... + b_{n-1} \cdot x^{n-1}$, design an algorithm to that outputs $A(x) \cdot B(x)$.

- We have to obtain the polynomial $C(x) = A(x) \cdot B(x)$
- $C(x)$ may be written as:
  $C(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + ... + c_{2n-2} \cdot x^{2n-2}$
- What is c_i in terms of coefficients of A and B?
  - $c_i = a_i \cdot b_0 + a_{i-1} \cdot b_1 + ... + a_0 \cdot b_i$
- The vector $(c_0, ..., c_{2n-2})$ is called the *convolution* of vectors $(a_0, ..., a_{n-1})$ and $(b_0, ..., b_{n-1})$.

Algorithm

`SimpleMultiply`$((a_0, ..., a_{n-1}), (b_0, ..., b_{n-1}))$

- For $i = 0$ to $2n - 2$
  - For $j = 0$ to i
    - $c_i \leftarrow c_i + a_j \cdot b_{i-j}$
- return$((c_0, c_1, ..., c_{2n-2}))$

- What is the running time of the above algorithm?

Algorithm

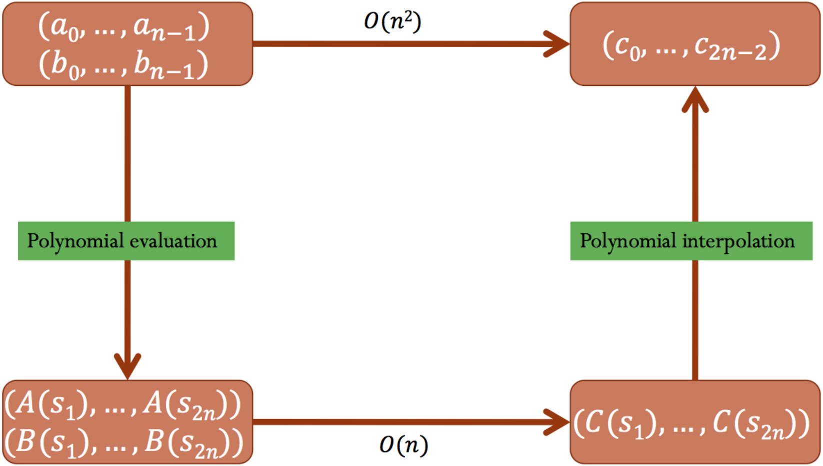
`SimpleMultiply`$((a_0, ..., a_{n-1}), (b_0, ..., b_{n-1}))$
- For $i = 0$ to $2n - 2$
  - For $j = 0$ to i
    - $c_i \leftarrow c_i + a_j \cdot b_{i-j}$
- return$((c_0, c_1, ..., c_{2n-2}))$

- What is the running time of the above algorithm? $O(n^2)$
- Is there another way to compute the polynomial $C(x)$?

- Another way to compute the polynomial $C(x)$:
  - Compute $A(s_1), A(s_2), ..., A(s_{2n})$.
  - Compute $B(s_1), B(s_2), ..., B(s_{2n})$.
  - Compute:
    - $C(s_1) = A(s_1) \cdot B(s_1)$
    - $C(s_2) = A(s_2) \cdot B(s_2)$
    - $\vdots$
    - $C(s_{2n}) = A(s_{2n}) \cdot B(s_{2n})$
  - Interpolate to obtain the polynomial $C(x)$.

- Another way to compute the polynomial $C(x)$:
  - Compute $A(s_1), A(s_2), ..., A(s_{2n})$.
  - Compute $B(s_1), B(s_2), ..., B(s_{2n})$.
  - Compute:
    - $C(s_1) = A(s_1) \cdot B(s_1)$
    - $C(s_2) = A(s_2) \cdot B(s_2)$
    - $\vdots$
    - $C(s_{2n}) = A(s_{2n}) \cdot B(s_{2n})$
  - Interpolate to obtain the polynomial $C(x)$.
- How fast can you compute $A(s)$ given value of s?

- Another way to compute the polynomial $C(x)$:
  - Compute $A(s_1), A(s_2), ..., A(s_{2n})$.
  - Compute $B(s_1), B(s_2), ..., B(s_{2n})$.
  - Compute:
    - $C(s_1) = A(s_1) \cdot B(s_1)$
    - $C(s_2) = A(s_2) \cdot B(s_2)$
    - $\vdots$
    - $C(s_{2n}) = A(s_{2n}) \cdot B(s_{2n})$
  - Interpolate to obtain the polynomial $C(x)$.
- How fast can you compute $A(s)$ given value of s?
  - $O(n)$ arithmetic operations using *Horner's rule*.
  - $A(s) = a_0 + s \cdot (a_1 + s \cdot (a_2 + ... + s \cdot (a_{n-1}) ...))$


$(a_0, \ldots, a_{n-1})$
$(b_0, \ldots, b_{n-1})$
$O(n^2)$
$(c_0, \ldots, c_{2n-2})$
Polynomial evaluation
Polynomial interpolation
$(A(s_1), \ldots, A(s_{2n}))$
$(B(s_1), \ldots, B(s_{2n}))$
$O(n)$
$(C(s_1), \ldots, C(s_{2n}))$

- Polynomial interpolation: We have $C(s_1), ..., C(s_{2n})$ and we need to compute $(c_0, ..., c_{2n-2})$.

$$\begin{pmatrix} 1 & s_1 & (s_1)^2 & \dots & (s_1)^{2n-1} \\ 1 & s_2 & (s_2)^2 & \dots & (s_2)^{2n-1} \\ 1 & s_3 & (s_3)^2 & \dots & (s_3)^{2n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & s_{2n} & (s_{2n})^2 & \dots & (s_{2n})^{2n-1} \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{2n-1} \end{pmatrix} = \begin{pmatrix} C(s_1) \\ C(s_2) \\ C(s_3) \\ \vdots \\ C(s_{2n}) \end{pmatrix}$$

- Is the above square matrix *invertible*?

- Polynomial interpolation: We have $C(s_1), ..., C(s_{2n})$ and we need to compute $(c_0, ..., c_{2n-2})$.

$$\begin{pmatrix} 1 & s_1 & (s_1)^2 & \dots & (s_1)^{2n-1} \\ 1 & s_2 & (s_2)^2 & \dots & (s_2)^{2n-1} \\ 1 & s_3 & (s_3)^2 & \dots & (s_3)^{2n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & s_{2n} & (s_{2n})^2 & \dots & (s_{2n})^{2n-1} \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{2n-1} \end{pmatrix} = \begin{pmatrix} C(s_1) \\ C(s_2) \\ C(s_3) \\ \vdots \\ C(s_{2n}) \end{pmatrix}$$

- Is the above square matrix *invertible*?
- Fact: A square matrix is invertible iff its determinant is non-zero.

- Polynomial interpolation: We have $C(s_1), ..., C(s_{2n})$ and we need to compute $(c_0, ..., c_{2n-2})$.

$$\begin{pmatrix} 1 & s_1 & (s_1)^2 & \dots & (s_1)^{2n-1} \\ 1 & s_2 & (s_2)^2 & \dots & (s_2)^{2n-1} \\ 1 & s_3 & (s_3)^2 & \dots & (s_3)^{2n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & s_{2n} & (s_{2n})^2 & \dots & (s_{2n})^{2n-1} \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{2n-1} \end{pmatrix} = \begin{pmatrix} C(s_1) \\ C(s_2) \\ C(s_3) \\ \vdots \\ C(s_{2n}) \end{pmatrix}$$

- Is the above square matrix *invertible*?
- Fact: A square matrix is invertible iff its determinant is non-zero.
- The square matrix above has a special name: *Vandermonde* matrix.

- Fact: A square matrix is invertible iff its determinant is non-zero.
- The square matrix above has a special name: *Vandermonde* matrix.
- Claim 1: For any Vandermonde matrix V shown below,

$$V = \begin{pmatrix} 1 & s_1 & (s_1)^2 & \dots & (s_1)^{2n-1} \\ 1 & s_2 & (s_2)^2 & \dots & (s_2)^{2n-1} \\ 1 & s_3 & (s_3)^2 & \dots & (s_3)^{2n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & s_{2n} & (s_{2n})^2 & \dots & (s_{2n})^{2n-1} \end{pmatrix}$$

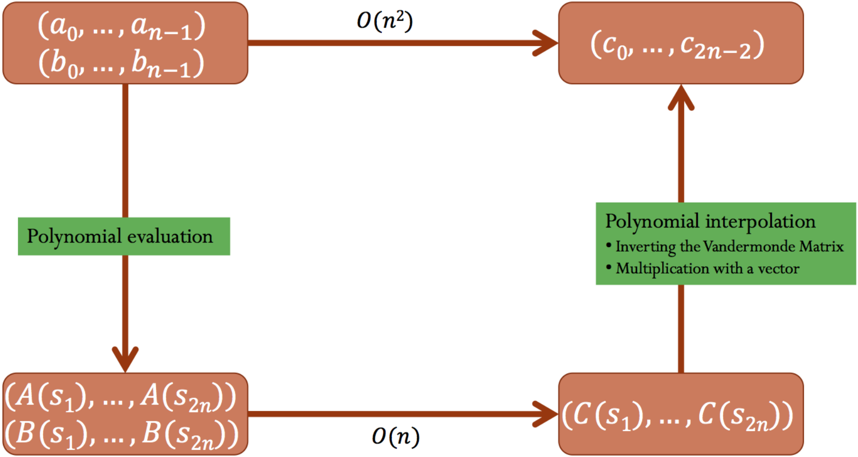
$Det(V) = \prod_{1 \leq j < i \leq 2n}(s_i - s_j)$.

- Fact: A square matrix is invertible iff its determinant is non-zero.
- The square matrix above has a special name: *Vandermonde* matrix.
- Claim 1: For any Vandermonde matrix V shown below,

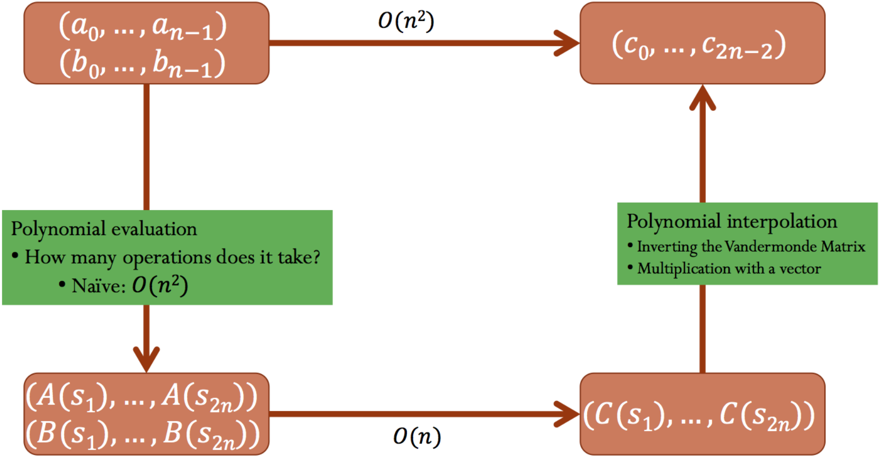
$$V = \begin{pmatrix} 1 & s_1 & (s_1)^2 & \dots & (s_1)^{2n-1} \\ 1 & s_2 & (s_2)^2 & \dots & (s_2)^{2n-1} \\ 1 & s_3 & (s_3)^2 & \dots & (s_3)^{2n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & s_{2n} & (s_{2n})^2 & \dots & (s_{2n})^{2n-1} \end{pmatrix}$$

$Det(V) = \prod_{1 \leq j < i \leq 2n} (s_i - s_j)$.

- So, as long as we use distict values of $s_1, ..., s_{2n}$, we will be able to do polynomial interpolation.


$(a_0, \ldots, a_{n-1})$
$(b_0, \ldots, b_{n-1})$
$O(n^2)$
$(c_0, \ldots, c_{2n-2})$
Polynomial evaluation
Polynomial interpolation
• Inverting the Vandermonde Matrix
• Multiplication with a vector
$(A(s_1), \ldots, A(s_{2n}))$
$(B(s_1), \ldots, B(s_{2n}))$
$O(n)$
$(C(s_1), \ldots, C(s_{2n}))$

Divide and Conquer
Fast Fourier Transform



- Example of polynomial evaluation:
  - $A(x) = 3 + 4x + 6x^2 + 2x^3 + x^4 + 10x^5 + 2x^6 + x^7$
  - $A(x) = (3 + 6x^2 + x^4 + 2x^6) + x \cdot (4 + 2x^2 + 10x^4 + x^6)$
  - Let $A_0(x) = 3 + 6x^2 + x^4 + 2x^6$
  - Let $A_1(x) = 4 + 2x^2 + 10x^4 + x^6$
  - How do we compute $A(1)$?
    - $A_0(1) = 12$, $A_1(1) = 17$.
    - So, $A(1) = A_0(1) + 1 \cdot A_1(1) = 12 + 17 = 29$.
  - Now, suppose we want to compute $A(-1)$.

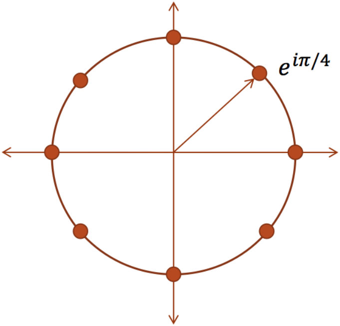
$$\begin{aligned} A(-1) &= A_0(-1) + (-1) \cdot A_1(-1) \\ &= A_0(1) + (-1) \cdot A_1(1) \\ &= 12 - 17 = -5 \end{aligned}$$

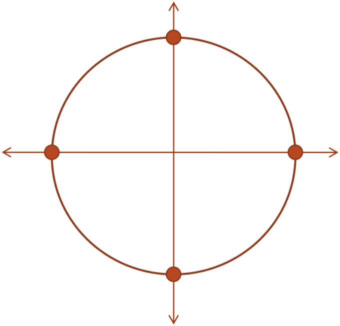
- Example of polynomial evaluation:
  - $A(x) = 3 + 4x + 6x^2 + 2x^3 + x^4 + 10x^5 + 2x^6 + x^7$
  - $A(x) = (3 + 6x^2 + x^4 + 2x^6) + x \cdot (4 + 2x^2 + 10x^4 + x^6)$
  - Let $A_0(x) = 3 + 6x^2 + x^4 + 2x^6$
  - Let $A_1(x) = 4 + 2x^2 + 10x^4 + x^6$
  - How do we compute $A(1)$?
    - $A_0(1) = 12$, $A_1(1) = 17$.
    - So, $A(1) = A_0(1) + 1 \cdot A_1(1) = 12 + 17 = 29$.
  - Now, suppose we want to compute $A(-1)$.

$$\begin{aligned} A(-1) &= A_0(-1) + (-1) \cdot A_1(-1) \\ &= A_0(1) + (-1) \cdot A_1(1) \\ &= 12 - 17 = -5 \end{aligned}$$

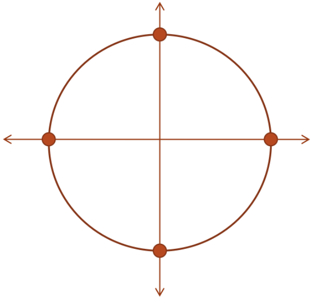
  - If we want to compute A on $-1, 1, -2, 2, -3, 3, -4, 4$, then we only need to compute A_0 and A_1 on $1, 2, 3, 4$.

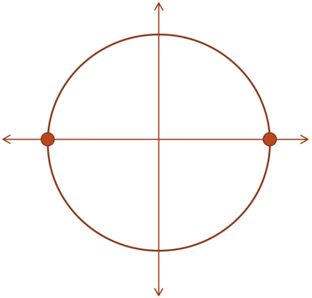
- Example of polynomial evaluation:
  - $A(x) = 3 + 4x + 6x^2 + 2x^3 + x^4 + 10x^5 + 2x^6 + x^7$
  - $A(x) = (3 + 6x^2 + x^4 + 2x^6) + x \cdot (4 + 2x^2 + 10x^4 + x^6)$
  - Let $A_0(x) = 3 + 6x^2 + x^4 + 2x^6$
  - Let $A_1(x) = 4 + 2x^2 + 10x^4 + x^6$



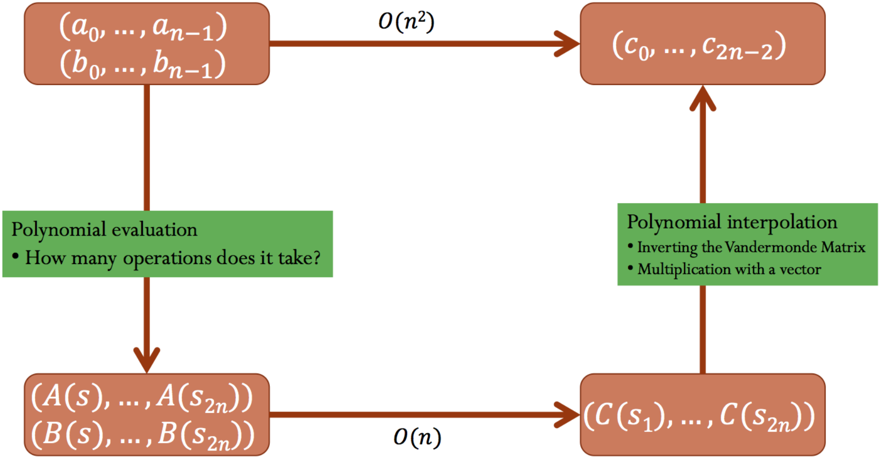


- Example of polynomial evaluation:
  - $A(x) = 3 + 4x + 6x^2 + 2x^3 + x^4 + 10x^5 + 2x^6 + x^7$
  - $A(x) = (3 + 6x^2 + x^4 + 2x^6) + x \cdot (4 + 2x^2 + 10x^4 + x^6)$
  - Let $A_0(x) = 3 + 6x^2 + x^4 + 2x^6$
  - Let $A_1(x) = 4 + 2x^2 + 10x^4 + x^6$
  - Let $A_{00}(x) = 3 + x^4$, $A_{01}(x) = 6 + 2x^4$
  - Let $A_{10}(x) = 4 + 10x^4$, $A_{11}(x) = 2 + x^4$





Divide and Conquer
Fast Fourier Transform



- Can we choose $s_1, ..., s_{2n}$ in a more clever manner so that evaluating the polynomials A and B on these points cost fewer operations?
- We will use complex roots of unity!
  - We will use $2n$ roots of the equation

$$x^{2n} - 1 = 0$$

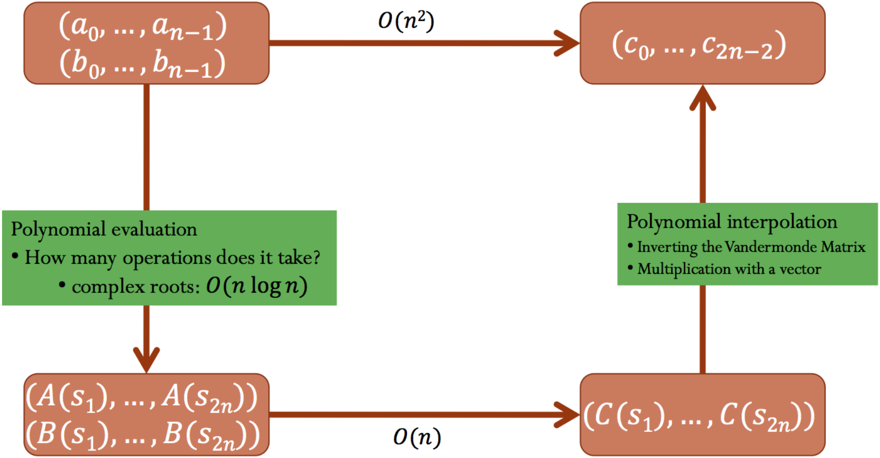
  - $s_1 = e^{1 \cdot \frac{2\pi i}{2n}}$
  - $s_2 = e^{2 \cdot \frac{2\pi i}{2n}}$
  - $\vdots$
  - $s_j = e^{j \cdot \frac{2\pi i}{2n}}$
  - $\vdots$

- Let w be one of the $2n$ roots of unity
- $A(w) = (a_0 + a_2 w^2 + a_4 w^4 + ...) + w \cdot (a_1 + a_3 w^2 + a_5 w^4 + ...)$
- $A(w) = A_0(w^2) + w \cdot A_1(w^2)$
- If we have $A_0(w^2)$ and $A_1(w^2)$, then computing $A(w)$ takes a constant number of operations.
- Suppose $T(n)$ denotes the worst case time to compute a polynomial at all $2n$ roots of unity.
- Using the above equation, we can say that:

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

  - Since w^2 is one of the n^{th} roots of unity.

Divide and Conquer
Fast Fourier Transform



- Claim 2: Let $w = e^{\frac{2\pi i}{2n}}$. Let V be the Vandermonde matrix w.r.t. the $2n$ roots of unity. That is,

$$V = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & (w^1)^1 & (w^1)^2 & \dots & (w^1)^{2n-1} \\ 1 & (w^2)^1 & (w^2)^2 & \dots & (w^2)^{2n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & (w^{2n-1})^1 & (w^{2n-1})^2 & \dots & (w^{2n-1})^{2n-1} \end{pmatrix}$$

Then $[V^{-1}]_{ij} = \frac{w^{-ij}}{2n}$. That is,

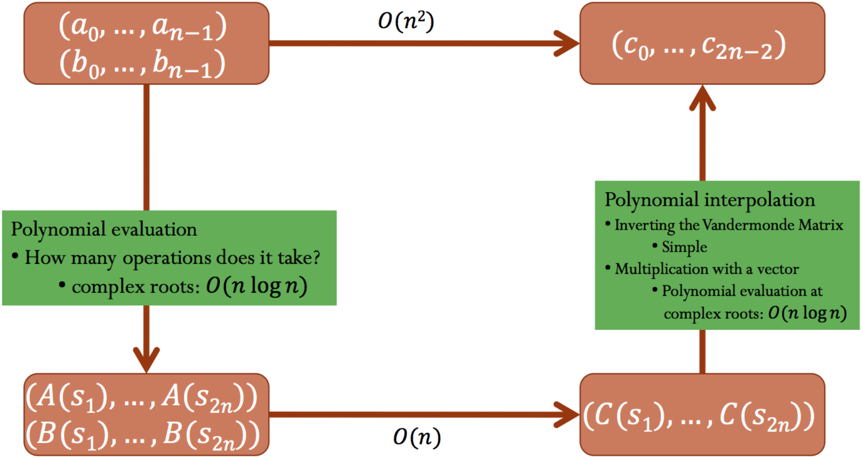
$$V^{-1} = \frac{1}{2n} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & (w^{-1})^1 & (w^{-1})^2 & \dots & (w^{-1})^{2n-1} \\ 1 & (w^{-2})^1 & (w^{-2})^2 & \dots & (w^{-2})^{2n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & (w^{-(2n-1)})^1 & (w^{-(2n-1)})^2 & \dots & (w^{-(2n-1)})^{2n-1} \end{pmatrix}$$

- We have

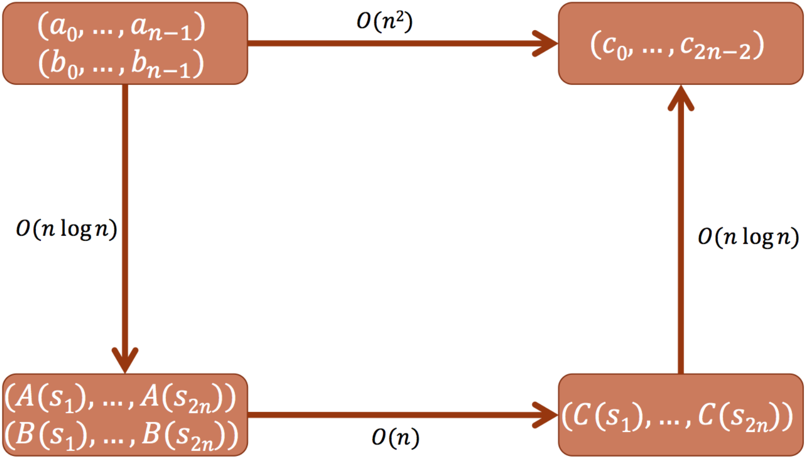
1. $$V^{-1} = \frac{1}{2n} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & (w^{-1})^1 & (w^{-1})^2 & \dots & (w^{-1})^{2n-1} \\ 1 & (w^{-2})^1 & (w^{-2})^2 & \dots & (w^{-2})^{2n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & (w^{-(2n-1)})^1 & (w^{-(2n-1)})^2 & \dots & (w^{-(2n-1)})^{2n-1} \end{pmatrix},$$

2. $$V \cdot \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2n-1} \end{pmatrix} = \begin{pmatrix} C(1) \\ C(w) \\ \vdots \\ C(w^{2n-1}) \end{pmatrix}$$

- How do we compute c_i's?



$(a_0, \dots, a_{n-1})$
$(b_0, \dots, b_{n-1})$
$O(n^2)$
$(c_0, \dots, c_{2n-2})$
Polynomial evaluation
• How many operations does it take?
• complex roots: $O(n \log n)$
Polynomial interpolation
• Inverting the Vandermonde Matrix
• Simple
• Multiplication with a vector
• Polynomial evaluation at complex roots: $O(n \log n)$
$(A(s_1), \dots, A(s_{2n}))$
$(B(s_1), \dots, B(s_{2n}))$
$O(n)$
$(C(s_1), \dots, C(s_{2n}))$


$(a_0, \ldots, a_{n-1})$
$(b_0, \ldots, b_{n-1})$
$O(n^2)$
$(c_0, \ldots, c_{2n-2})$
$O(n \log n)$
$O(n \log n)$
$(A(s_1), \ldots, A(s_{2n}))$
$(B(s_1), \ldots, B(s_{2n}))$
$O(n)$
$(C(s_1), \ldots, C(s_{2n}))$

End