Fast Integer Multiplication

By

Sai Pranitha Kambham

Ayesha Anjum Shaik

Sandeep Pothineni

Problem: Multiply two n-bit integers really fast.

Familiar Algorithms

- **Grade School Algorithm**
  - Time Complexity – $O(n^2)$
- **Karatsuba's Algorithm**
  - $T(n) = 3T\left(\frac{n}{2}\right) + \theta(n)$
  - Time Complexity $-O\left(n^{log_2^3}\right) = O(n^{1.585})$

Multiplication is a sub-routine in many algorithms

- Assume multiplication is done in $P(n)$ time, then
  - Division, modulo can be done in $O(P(n))$ time
  - Square-root in $O(P(n))$ time
  - GCD in $O(P(n)logn)$
  - n digits of π in $O(P(n)logn)$
  - Primality testing in $O(P(n)n)$

Schönhage–Strassen high level plan

- Multiplying integers reduces to multiplying polynomials with integer coefficients.
- Multiplying polynomials is easy in the “Values Representation”

Polynomials in their coefficient form

Now let us compute the time complexity of polynomial multiplication.

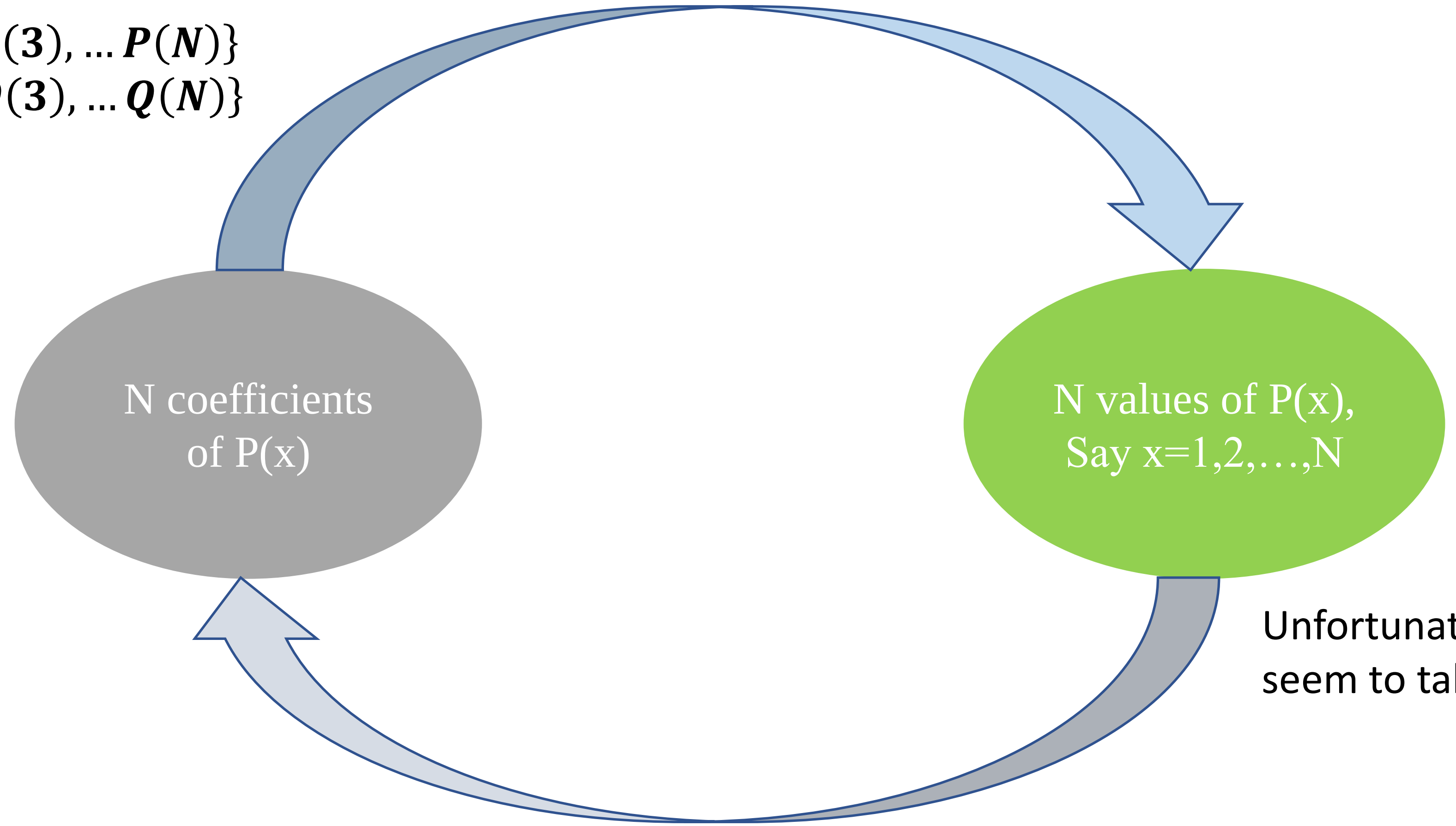
Let $P(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{N-1}x^{N-1}$

$Q(x) = b_0 + b_1x + b_2x^2 + \ldots + b_{N-1}x^{N-1}$

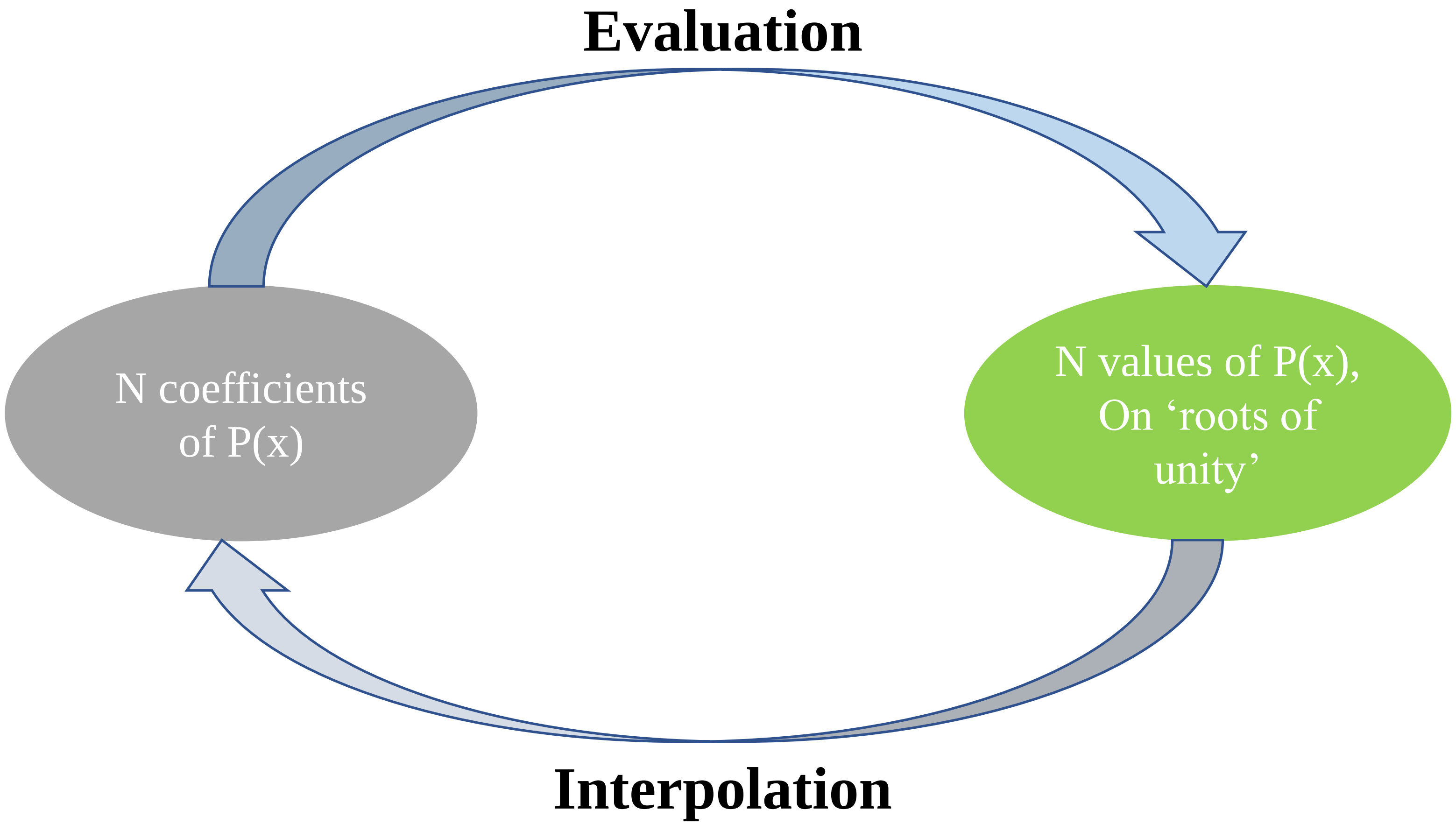
Multiplication of each of the coefficients takes $O(N^2)$time

Representation of polynomial in samples

$$\boldsymbol{P(x) = \{P(1), P(2), P(3), \ldots P(N)\}}$$
$$\boldsymbol{Q(x) = \{Q(1), Q(2), Q(3), \ldots Q(N)\}}$$



Unfortunately, these seem to take $O(N^2)$


Evaluation
N coefficients of P(x)
N values of P(x), On ‘roots of unity’
Interpolation

Discrete and Inverse discrete Fourier Transform

Let N be a power of 2

$S_N = \{1, \omega_N^1, \omega_N^2, \omega_N^3, \ldots, \omega_N^{N-1}\}$ is the set of N "complex roots of unity"

Let $P(x)$ be a polynomial of degree N-1

$$\text{P's coefficients} \xrightarrow[\text{Evaluation}]{\text{DFT}_\text{N}} \text{P's values on } S_N$$

$$\text{P's values on } S_N \xrightarrow[\text{Interpolation}]{\text{IDFT}_\text{N}} \text{P's coefficients}$$

Calculation of the time complexity on multiplying polynomials with the FFT

Input : $P(x)$ and $Q(x)$ be polynomials of degree < N

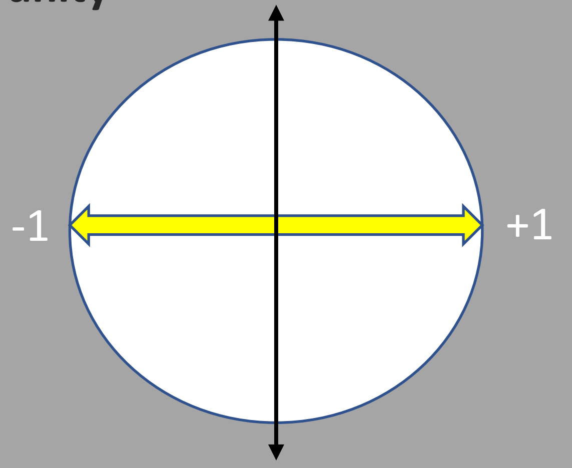
Output : $R(x) = P(x).Q(x)$ of degree 2N

- Use DFT_{2N} to get $P(w)$, $Q(w)$ $\forall$ $w \in S_{2N}$ --- $O(NlogN)$
- Multiply pairs, results in $R(w)$ $\forall$ $w \in S_{2N}$ --- $O(N)$
- Use $IDFT_{2N}$ to get R's coefficients --- $O(NlogN)$

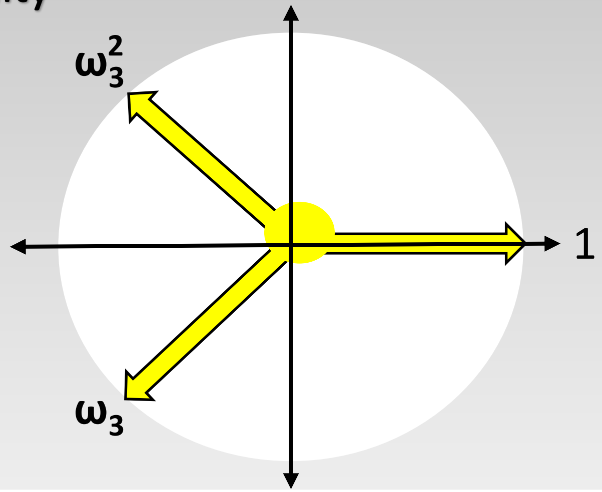
Hence the polynomial multiplication takes $O(NlogN)$ time

Roots Of Unity

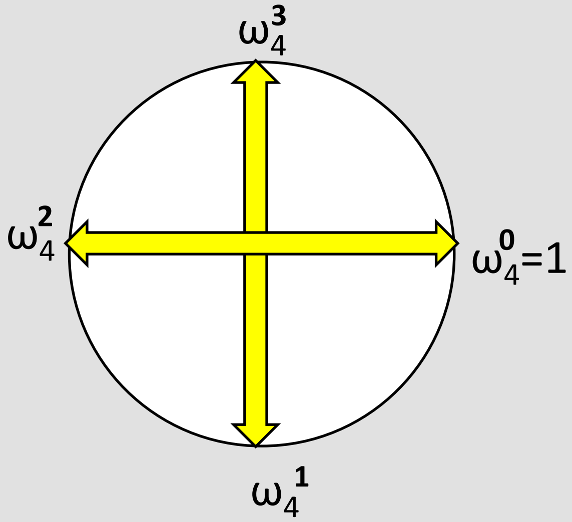
Square roots of unity



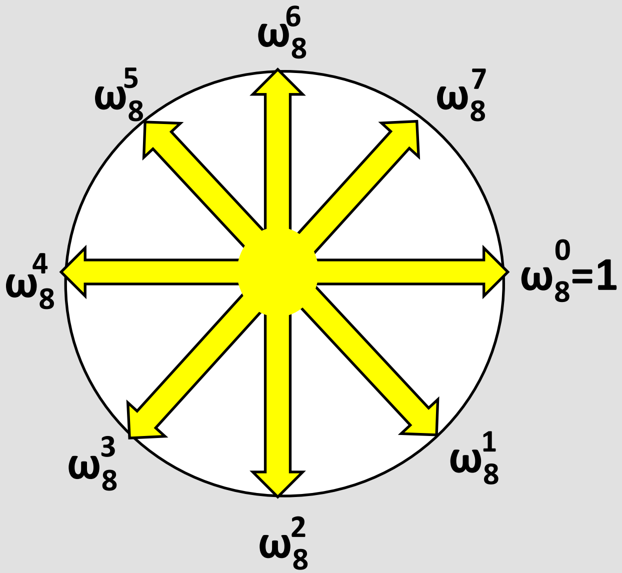
Cube roots of unity



Fourth roots of unity



8^{th} roots of unity



Evaluation in $\{1, \omega, \omega^2, \omega^3, \omega^4, \omega^5, \omega^6, \omega^7\}$

Say $P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega^8 & \omega^{10} & \omega^{12} & \omega^{14} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \omega^{12} & \omega^{15} & \omega^{18} & \omega^{21} \\ 1 & \omega^4 & \omega^8 & \omega^{12} & \omega^{16} & \omega^{20} & \omega^{24} & \omega^{28} \\ 1 & \omega^5 & \omega^{10} & \omega^{15} & \omega^{20} & \omega^{25} & \omega^{30} & \omega^{35} \\ 1 & \omega^6 & \omega^{12} & \omega^{18} & \omega^{24} & \omega^{30} & \omega^{36} & \omega^{42} \\ 1 & \omega^7 & \omega^{14} & \omega^{21} & \omega^{28} & \omega^{35} & \omega^{42} & \omega^{49} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} P(1) \\ P(\omega) \\ P(\omega^2) \\ P(\omega^3) \\ P(\omega^4) \\ P(\omega^5) \\ P(\omega^6) \\ P(\omega^7) \end{bmatrix}$$

Since $\omega^8 = 1$, all the exponents above can be reduced as

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} P(1) \\ P(\omega) \\ P(\omega^2) \\ P(\omega^3) \\ P(\omega^4) \\ P(\omega^5) \\ P(\omega^6) \\ P(\omega^7) \end{bmatrix}$$

$$DFT_8[j,k] = \omega^{jk \bmod 8} \quad (0 \leq \mathrm{j}, \mathrm{k} < 8)$$

Interpolation

$$\mathrm{DFT}_8 \bullet \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} P(1) \\ P(\omega) \\ P(\omega^2) \\ P(\omega^3) \\ P(\omega^4) \\ P(\omega^5) \\ P(\omega^6) \\ P(\omega^7) \end{bmatrix}$$

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \underbrace{\mathrm{DFT}_8^{-1}}_{\mathrm{IDFT}_8} \bullet \begin{bmatrix} P(1) \\ P(\omega) \\ P(\omega^2) \\ P(\omega^3) \\ P(\omega^4) \\ P(\omega^5) \\ P(\omega^6) \\ P(\omega^7) \end{bmatrix}$$

To generalize $IDFT_N[j,k] = \frac{1}{N}\omega^{-jk \bmod N}$

$$
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\
1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\
1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\
1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\
1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\
1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\
1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega
\end{bmatrix}
\bullet
\begin{bmatrix}
a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7
\end{bmatrix}
$$

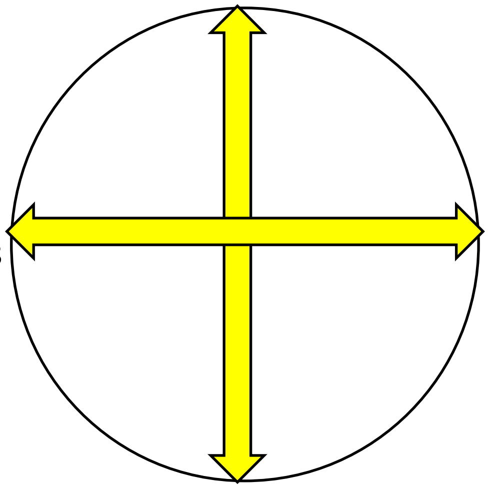
$$
= a_0 \bullet \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
+ a_1 \bullet \begin{bmatrix} 1 \\ \omega \\ \omega^2 \\ \omega^3 \\ \omega^4 \\ \omega^5 \\ \omega^6 \\ \omega^7 \end{bmatrix}
+ a_2 \bullet \begin{bmatrix} 1 \\ \omega^2 \\ \omega^4 \\ \omega^6 \\ 1 \\ \omega^2 \\ \omega^4 \\ \omega^6 \end{bmatrix}
+ a_3 \bullet \begin{bmatrix} 1 \\ \omega^3 \\ \omega^6 \\ \omega \\ \omega^4 \\ \omega^7 \\ \omega^2 \\ \omega^5 \end{bmatrix}
+ a_4 \bullet \begin{bmatrix} 1 \\ \omega^4 \\ 1 \\ \omega^4 \\ 1 \\ \omega^4 \\ 1 \\ \omega^4 \end{bmatrix}
+ a_5 \bullet \begin{bmatrix} 1 \\ \omega^5 \\ \omega^2 \\ \omega^7 \\ \omega^4 \\ \omega \\ \omega^6 \\ \omega^3 \end{bmatrix}
+ a_6 \bullet \begin{bmatrix} 1 \\ \omega^6 \\ \omega^4 \\ \omega^2 \\ 1 \\ \omega^6 \\ \omega^4 \\ \omega^2 \end{bmatrix}
+ a_7 \bullet \begin{bmatrix} 1 \\ \omega^7 \\ \omega^6 \\ \omega^5 \\ \omega^4 \\ \omega^3 \\ \omega^2 \\ \omega \end{bmatrix}
$$

$$= a_0 \bullet \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + a_2 \bullet \begin{bmatrix} 1 \\ \omega^2 \\ \omega^4 \\ \omega^6 \\ 1 \\ \omega^2 \\ \omega^4 \\ \omega^6 \end{bmatrix} + a_4 \bullet \begin{bmatrix} 1 \\ \omega^4 \\ 1 \\ \omega^4 \\ 1 \\ \omega^4 \\ 1 \\ \omega \end{bmatrix} + a_6 \bullet \begin{bmatrix} 1 \\ \omega^6 \\ \omega^4 \\ \omega^2 \\ 1 \\ \omega^6 \\ \omega^4 \\ \omega^2 \end{bmatrix}$$

$$DFT_4 \bullet \begin{bmatrix} a_0 \\ a_2 \\ a_4 \\ a_6 \end{bmatrix}$$

$$= \left(\quad \textbf{ditto} \quad \right)$$

$$+ a_1 \bullet \begin{bmatrix} 1 \\ \omega \\ \omega^2 \\ \omega^3 \\ \omega^4 \\ \omega^5 \\ \omega^6 \\ \omega^7 \end{bmatrix} + a_3 \bullet \begin{bmatrix} 1 \\ \omega^3 \\ \omega^6 \\ \omega \\ \omega^4 \\ \omega^7 \\ \omega^2 \\ \omega^5 \end{bmatrix} + a_5 \bullet \begin{bmatrix} 1 \\ \omega^5 \\ \omega^2 \\ \omega^7 \\ \omega^4 \\ \omega \\ \omega^6 \\ \omega^3 \end{bmatrix} + a_7 \bullet \begin{bmatrix} 1 \\ \omega^7 \\ \omega^6 \\ \omega^5 \\ \omega^4 \\ \omega^3 \\ \omega^2 \\ \omega \end{bmatrix}$$



$$= a_0 \bullet \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + a_2 \bullet \begin{bmatrix} 1 \\ \omega^2 \\ \omega^4 \\ \omega^6 \\ 1 \\ \omega^2 \\ \omega^4 \\ \omega^6 \end{bmatrix} + a_4 \bullet \begin{bmatrix} 1 \\ \omega^4 \\ 1 \\ \omega^4 \\ 1 \\ \omega^4 \\ 1 \\ \omega \end{bmatrix} + a_6 \bullet \begin{bmatrix} 1 \\ \omega^6 \\ \omega^4 \\ \omega^2 \\ 1 \\ \omega^6 \\ \omega^4 \\ \omega^2 \end{bmatrix}$$

$$+ a_1 \bullet \begin{bmatrix} 1 \\ \omega \\ \omega^2 \\ \omega^3 \\ \omega^4 \\ \omega^5 \\ \omega^6 \\ \omega^7 \end{bmatrix} + a_3 \bullet \begin{bmatrix} 1 \\ \omega^3 \\ \omega^6 \\ \omega \\ \omega^4 \\ \omega^7 \\ \omega^2 \\ \omega^5 \end{bmatrix} + a_5 \bullet \begin{bmatrix} 1 \\ \omega^5 \\ \omega^2 \\ \omega^7 \\ \omega^4 \\ \omega \\ \omega^6 \\ \omega^3 \end{bmatrix} + a_7 \bullet \begin{bmatrix} 1 \\ \omega^7 \\ \omega^6 \\ \omega^5 \\ \omega^4 \\ \omega^3 \\ \omega^2 \\ \omega \end{bmatrix}$$

$$= a_0 \bullet \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + a_2 \bullet \begin{bmatrix} 1 \\ \omega^2 \\ \omega^4 \\ \omega^6 \\ 1 \\ \omega^2 \\ \omega^4 \\ \omega^6 \end{bmatrix} + a_4 \bullet \begin{bmatrix} 1 \\ \omega^4 \\ 1 \\ \omega^4 \\ 1 \\ \omega^4 \\ 1 \\ \omega \end{bmatrix} + a_6 \bullet \begin{bmatrix} 1 \\ \omega^6 \\ \omega^4 \\ \omega^2 \\ 1 \\ \omega^6 \\ \omega^4 \\ \omega^2 \end{bmatrix}$$

Computable with 1 application of DFT_4 to (a_0, a_2, a_4, a_6), and repeating the same

$$+ a_1 \bullet \begin{bmatrix} 1 \\ \omega \\ \omega^2 \\ \omega^3 \\ \omega^4 \\ \omega^5 \\ \omega^6 \\ \omega^7 \end{bmatrix} + a_3 \bullet \begin{bmatrix} 1 \\ \omega^3 \\ \omega^6 \\ \omega \\ \omega^4 \\ \omega^7 \\ \omega^2 \\ \omega^5 \end{bmatrix} + a_5 \bullet \begin{bmatrix} 1 \\ \omega^5 \\ \omega^2 \\ \omega^7 \\ \omega^4 \\ \omega \\ \omega^6 \\ \omega^3 \end{bmatrix} + a_7 \bullet \begin{bmatrix} 1 \\ \omega^7 \\ \omega^6 \\ \omega^5 \\ \omega^4 \\ \omega^3 \\ \omega^2 \\ \omega \end{bmatrix}$$

Now to get this, apply the above to (a_1, a_3, a_5, a_7) and then multiply the j^{th} row with ω^j, for $0 \leq j < 8$

DFT_N reduces to 2 applications of $DFT_{N/2}$, plus O(N) additional operations.

$$T(N) = 2T\left(\frac{N}{2}\right) + O(N)$$

$$T(N) = O(N \log N)$$

Reduction to polynomial multiplication

- Given n-bit integers A,B, of base n

  i.e, let $l = \log n$, let $N = n/l$, and break the bits of A and B into N blocks of length l

$$A = a_{N-1}2^{(N-1)l} + \cdots + a_2 2^{2l} + a_1 2^l + a_0$$
$$B = b_{N-1}2^{(N-1)l} + \cdots + b_2 2^{2l} + b_1 2^l + b_0$$

$0 \leq a_i, b_i < n$, so they are Bounded Integers

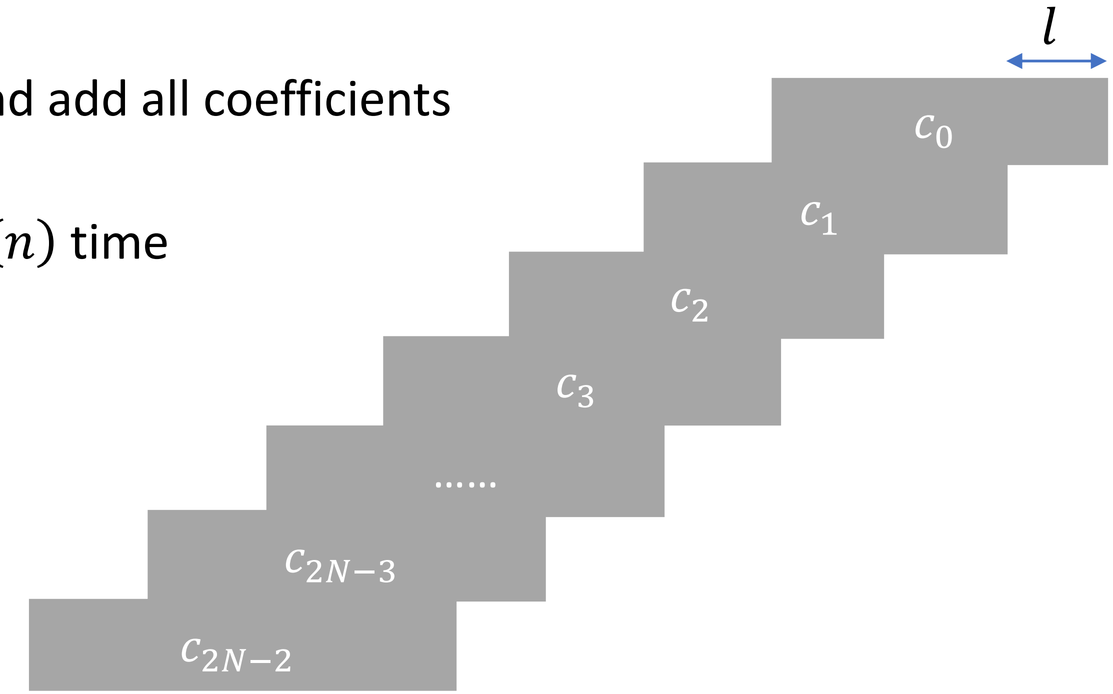
$$P(x) = a_{N-1}x^{N-1} + \cdots + a_2 x^2 + a_1 x + a_0$$
$$Q(x) = b_{N-1}x^{N-1} + \cdots + b_2 x^2 + b_1 x + b_0$$

and $R(x) = P(x).\,Q(x)$

Note that $A = P\left(2^l\right), B = Q(2^l)$, so $A.\,B = R(2^l)$

Shift l bits and add all coefficients

This takes $O(n)$ time



Each answer block is sum of ≤ 4 bounded integers

$$R(x) = c_{2N-2}x^{2N-2} + \cdots + c_2x^2 + c_1x + c_0$$

Conclusion:

Suppose we can multiply two polynomials of degree $< N$, with bounded integer coefficients in $T(N)$ time.

we can multiply two n-bit integers in $O(T\left(\frac{n}{\log(n)}\right))$ time

As $T(N) = O(NlogN)$

$O(\frac{n}{\log n}\log\frac{n}{\log n}) \cong O(n)$

Applications

- Great Internet Mersenne prime search
- Computing approximations of Π
- Areas in encryption
- Kronecker substitution

THANK YOU