Algoritmi avanzati
A.A. 2012-2013

Roberto Battiti
Paolo Campigotto

AVVERTENZA: lucidi da usare come ausilio mnemonico e lista degli argomenti svolti a lezione.

Non sostituiscono in alcun modo il libro di testo che va usato per lo studio approfondito.

Polynomials and FFT

- $A(x) = \Sigma_{j=0}^{n-1} a_j x^j$
- Straightforward addition $\Theta(n)$, multiplication $\Theta(n^2)$
- FFT (Fast Fourier Transform) reduces multiplication time to $\Theta(n \log n)$
- relevant application: signal processing: from time to frequency domain

Polynomials: basics

- $A(x) = \sum_{j=0}^{n-1} a_j x^j \quad B(x) = \sum_{j=0}^{n-1} b_j x^j$
- Coefficient, degree, degree-bound (greater)
- $C(x) = A(x)B(x) = B(x) = \sum_{j=0}^{2n-2} c_j x^j$

  $c_j = \sum_{k=0}^{j} a_k b_{j-k}$ convolution $c = a \otimes b$
- Coefficient representation, **evaluation** $\Theta(n)$

  Horner's rule

$$A(x) = a_0+x(a_1+x(a_2+\ldots. (a_{n-2}+a_{n-1}x))))$$

Polynomials: basics

- Point-value representation

(x_0,y_0) …. (x_{n-1}, y_{n-1})

- **Interpolation**: inverse of evaluation

Interpolation

- Uniqueness of interpolating polynomial

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

- Vandermonde matrix, det $\prod_{0 \le j < k \le n-1} (x_k - x_j)$

$$a = V(x_0, x_1, \ldots, x_{n-1})^{-1} y .$$

No. of points= number of parameters

- Faster with Lagrange formula $\Theta(n^2)$

$$A(x) = \sum_{k=0}^{n-1} y_k \frac{\prod_{j \ne k} (x - x_j)}{\prod_{j \ne k} (x_k - x_j)} .$$

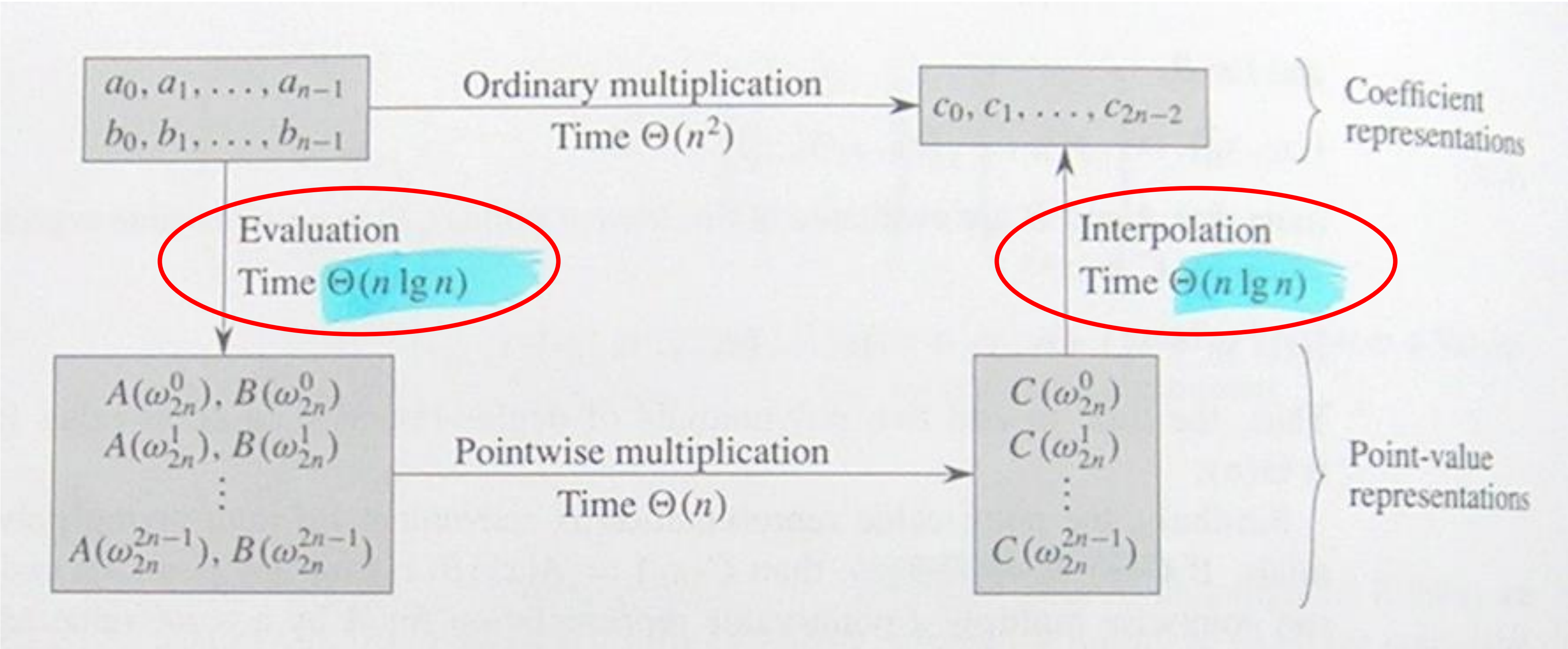
Evaluation and interpolation are well defined and $\Theta(n^2)$

Point-value and coefficient representation

- Point-value is convenient for multiplying polynomial… but need 2n pairs $\Theta(n)$
- Can I use this result for multiplying polynomials in coefficient representation?

  (without being killed by the $\Theta(n^2)$ evaluation and interpolation)

Fast multiplication w. coefficient representation



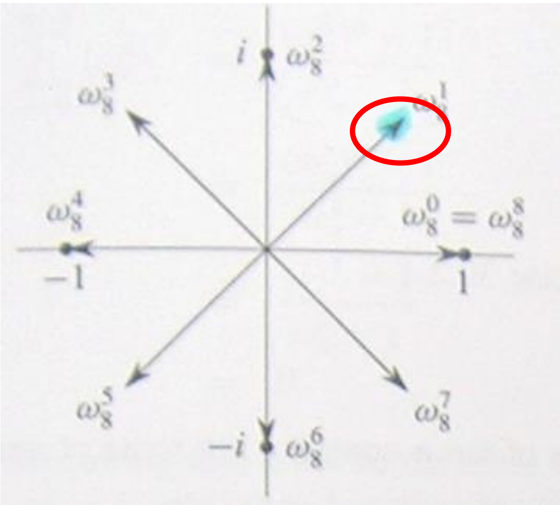
- Double degree bound, evaluate, point-wise multiply, interpolate

Fast multiplication w. coefficient representation

- Use “special points”: complex roots of unity!
- $\omega_n = e^{2\pi i/n}$ principal nth root
- $\omega_n^0\ \omega_n^1\ \omega_n^2 \ldots \omega_n^{n-1}$
- Cancellation lemma

$\omega_{dn}^{dk} = \omega_n^k$

$e^{iu} = \cos(u) + i\ \mathrm{sen}(u)$



- Halving lemma

n even:square nth roots

→ get n/2 rooths

- Summation lemma (k not divisible by n)

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

Fast multiplication w. coefficient representation

- Summation lemma (k not divisible by n)

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

$$
\begin{aligned}
\sum_{j=0}^{n-1} (\omega_n^k)^j &= \frac{(\omega_n^k)^n - 1}{\omega_n^k - 1} \\
&= \frac{(\omega_n^n)^k - 1}{\omega_n^k - 1} \\
&= \frac{(1)^k - 1}{\omega_n^k - 1} \\
&= 0 .
\end{aligned}
$$

DFT (discrete Fourier transform)

n is power of 2

Cooley, James W., and John W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput. ***19****, 297–301 (1965).*

$$A(x) = \sum_{j=0}^{n-1} a_j x^j$$

$$\begin{aligned} y_k &= A(\omega_n^k) \\ &= \sum_{j=0}^{n-1} a_j \omega_n^{kj} . \end{aligned}$$

$$y = \mathrm{DFT}_n(a)$$

Use *divide et impera* (divide and conquer) approach

DFT

$$
\begin{aligned}
A^{[0]}(x) &= a_0 + a_2x + a_4x^2 + \cdots + a_{n-2}x^{n/2-1} , \\
A^{[1]}(x) &= a_1 + a_3x + a_5x^2 + \cdots + a_{n-1}x^{n/2-1} .
\end{aligned}
$$

Note that $A^{[0]}$ contains all the even-index coefficients of A (the binary representation of the index ends in 0) and $A^{[1]}$ contains all the odd-index coefficients (the binary representation of the index ends in 1). It follows that

$$A(x) = A^{[0]}(x^2) + xA^{[1]}(x^2) , \tag{30.9}$$

so that the problem of evaluating $A(x)$ at $\omega_n^0, \omega_n^1, \ldots, \omega_n^{n-1}$ reduces to

1. evaluating the degree-bound $n/2$ polynomials $A^{[0]}(x)$ and $A^{[1]}(x)$ at the points

   $$(\omega_n^0)^2, (\omega_n^1)^2, \ldots, (\omega_n^{n-1})^2 ,$$

   and then

2. combining the results according to equation (30.9).

Θ(n log n)

$$
\begin{aligned}
T(n) &= 2T(n/2) + \Theta(n) \\
&= \Theta(n \lg n) .
\end{aligned}
$$

DFT

```
RECURSIVE-FFT(a)
n ← length[a]                          ▷ n is a power of 2.
if n = 1
    then return a
ω_n ← e^{2πi/n}
ω ← 1
a^[0] ← (a_0, a_2, ..., a_{n-2})
a^[1] ← (a_1, a_3, ..., a_{n-1})
y^[0] ← RECURSIVE-FFT(a^[0])
y^[1] ← RECURSIVE-FFT(a^[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_n
return y                               ▷ y is assumed to be a column vector.
```

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

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

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

Lines 11–12 combine the results of the recursive $\mathrm{DFT}_{n/2}$ calculations. For $y_0, y_1, \ldots, y_{n/2-1}$, line 11 yields

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

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

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

Interpolationg at the complex roots of unity: inverse DFT

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

(k, j) entry of V_n is ω_n^{kj},

Theorem 30.7
For $j, k = 0, 1, \ldots, n-1$, the (j, k) entry of V_n^{-1} is ω_n^{-kj}/n.

Proof We show that $V_n^{-1}V_n = I_n$, the $n \times n$ identity matrix. Consider the (j, j') entry of $V_n^{-1}V_n$:

$$\begin{aligned} [V_n^{-1}V_n]_{jj'} &= \sum_{k=0}^{n-1}(\omega_n^{-kj}/n)(\omega_n^{kj'}) \\ &= \sum_{k=0}^{n-1}\omega_n^{k(j'-j)}/n . \end{aligned}$$

Inverse DFT

$$a_j = \frac{1}{n}\sum_{k=0}^{n-1} y_k \omega_n^{-kj}$$

- Similar to:

$$\begin{aligned} y_k &= A(\omega_n^k) \\ &= \sum_{j=0}^{n-1} a_j \omega_n^{kj} . \end{aligned}$$

….also $\boxed{\Theta(n \log n)}$

- Convolution theorem (n pow. of 2, padding with 0)

$$a \otimes b = \mathrm{DFT}_{2n}^{-1}(\mathrm{DFT}_{2n}(a) \cdot \mathrm{DFT}_{2n}(b))$$