Representation of polynomials

3. Point-value representation

Interpolation lemma:

Any polynomial p(x)∈ *R[x]* of degree *n* is uniquely defined by $n+1$ pairs $(x_i, p(x_i))$, where $i = 0,\dots,n$ and $x_i \neq x_j$ for $i \neq j$.

point value

Example:

The polynomial

$$p(x) = 3x(x-2)(x-3)$$

is uniquely defined by the point-value pairs (0,0), (1,6), (2,0), (3,0).

Operations on polynomials

$p, q \in R[x]$, degree(p) = degree(q) = n

- **Coefficient representation**

  Addition: $O(n)$

  Multiplication: $O(n^2)$ $\longrightarrow$ $O(n \log n)$ FFT

  Evaluation at x_0: $O(n)$ Horner

- **Point-value representation**

$$p = (x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

$$q = (x_0, z_0), (x_1, z_1), \ldots, (x_n, z_n)$$

Operations on polynomials

Addition:

$$p + q = (x_0, \underline{y_0 + z_0}), (x_1, \underline{y_1 + z_1}), \ldots, (x_n, \underline{y_n + z_n})$$

Running time: O(*n*)

Multiplication:

$$p \cdot q = (x_0, \underline{y_0 \cdot z_0}), (x_1, \underline{y_1 \cdot z_1}), \ldots, (x_n, \underline{y_n \cdot z_n})$$

(Condition: $n \geq$ degree(pq))

Running time: O(n)

Evaluation at point *x´:* ??

Convert polynomial to coefficient representation (interpolation)

Polynomial multiplication

Compute the product of two polynomials *p, q* of degree < *n*:

Initially coefficient representation

$\deg(pq) \leq 2n-2$, $2n-1$ points suffice

p,q of degree *n-1*, *n* coefficients

we evaluate at $2n$ points.
even number of points useful for D&C.

Evaluation: $x_0, x_1, \ldots, x_{2n-1}$

$n \cdot$ Horner : $O(n^2)$
FFT : $O(n \log n)$

$2n$ point-value pairs $(x_i, p(x_i))$ und $(x_i, q(x_i))$

Pointwise multiplication $O(n)$

$2n$ point-value pairs $(x_i, pq(x_i))$

Interpolation ? $O(n \log n)$

pq of degree *2n-2*, *2n-1* coefficients

Divide-and-conquer approach

for polynomial evaluation

Idea: (assume n is even)

$$p(x) = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1}$$

$$= a_0 + a_2 x^2 + \ldots + a_{n-2} x^{n-2} +$$

$$a_1 x + a_3 x^3 + \ldots + a_{n-1} x^{n-1}$$

$$a_4 \cdot (x^2)^2 + a_6 (x^2)^3 + \ldots$$

$$= a_0 + a_2 x^2 + \ldots + a_{n-2} \left(x^2\right)^{(n-2)/2} +$$

$$a_5 (x^2)^2 + a_7 (x^2)^3 + \ldots$$

$$x \left(a_1 + a_3 x^2 + \ldots + a_{n-1} \left(x^2\right)^{(n-2)/2} \right)$$

$$= p_0\left(x^2\right) + x p_1\left(x^2\right)$$

$$a_4 \cdot x^2 + \ldots$$

$\rightarrow$ $p_0(x) = a_0 + a_2 x + \ldots + a_{n-2} x^{(n-2)/2}$

$\rightarrow$ $p_1(x) = a_1 + a_3 x + \ldots + a_{n-1} x^{(n-2)/2}$

Select $x_0, \ldots, x_{2n-1}$ such that the computations of $p(x_k)$ and $p(x_{k+n})$ are almost identical.

Idea: Instead of eval. one poly with degree n evaluate two poly. with degree $n/2$

Goal: Evaluate $2n$ points with running time $O(n \log n)$

Representation of *p(x)*

i: imaginary unit of complex numbers

Recall: $e^{x \cdot i} = \cos x + i \cdot \sin x$

Assume: degree(*p*) < *n*

3a. Values of the *n* powers of the principal *n*th root of unity

Def: $\omega_n := e^{2\pi i/n} = \cos 2\pi/n + i \sin 2\pi/n$

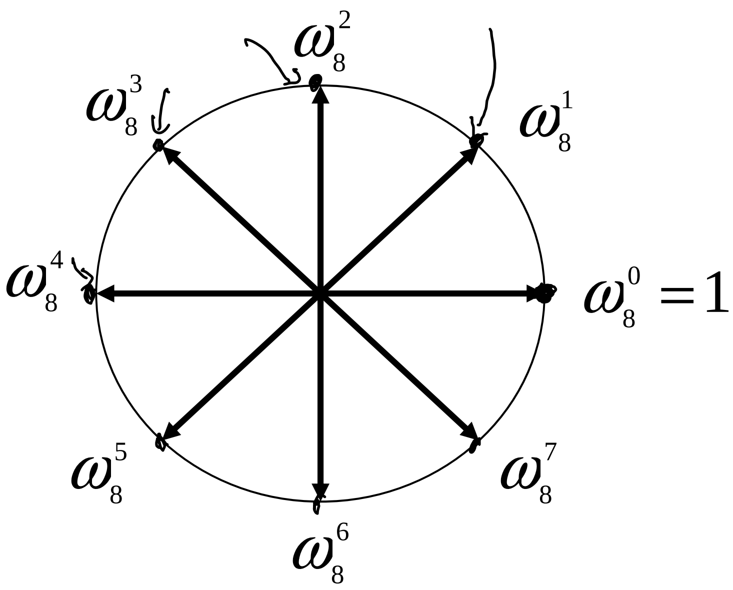
$i = \sqrt{-1} \qquad e^{2\pi i} = 1$

Powers of ω_n (roots of unity):

$$1 = \omega_n^0, \omega_n^1, \ldots, \omega_n^{n-1}$$

$$\omega_n^k = \left(e^{2\pi i/n}\right)^k = e^{\frac{2\pi k}{n} \cdot i}$$

$$= \cos \frac{2\pi k}{n} + i \sin \frac{2\pi k}{n}$$



Discrete Fourier Transform

The values $p(\omega_n^i)$ of the n powers of ω_n uniquely define p if degree(p) < n.

Discrete Fourier Transform (DFT)

$$DFT_n(p) = \left(p(\omega_n^0), p(\omega_n^1), \ldots, p(\omega_n^{n-1})\right)$$

Evaluate p at the points $\omega_n^0, \omega_n^1, \ldots, \omega_n^{n-1}$ and gather the results in a vector

Example: $n = 4$

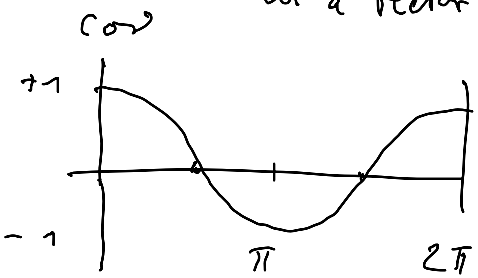
$$e^{ix} = \cos x + i \sin x$$

$$\omega_4^0 = e^{0i} = \cos(0) + i\sin(0) = 1$$

$$\omega_4^1 = e^{\cancel{2}\pi i/\cancel{4}2} = \cos(\pi/2) + i\sin(\pi/2) = i$$

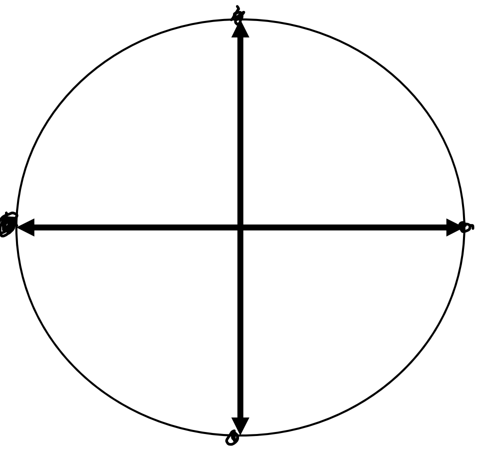
$$\omega_4^2 = \left(e^{2\pi i/4}\right)^2 = \cos\pi + i\sin\pi = -1$$

$$\omega_4^3 = \left(e^{2\pi i/4}\right)^3 = \cos(3\pi/2) + i\sin(3\pi/2) = -i$$



Evaluation at the roots of unity

$$\omega_4^1 = i$$



$$\omega_2^1 = \omega_4^2 = -1 \qquad \omega_4^0 = 1$$

$$\omega_4^3 = -i$$

Evaluation at the roots of unity

$$p(x) = 3x^3 - 15x^2 + 18x$$

$$(\omega_4^0, p(\omega_4^0)) = (1, p(1)) = (1,6)$$

$$(\omega_4^1, p(\omega_4^1)) = (i, p(i)) = (i, 15 + 15i)$$

$$(\omega_4^2, p(\omega_4^2)) = (-1, p(-1)) = (-1, -36)$$

$$(\omega_4^3, p(\omega_4^3)) = (-i, p(-i)) = (-i, 15 - 15i)$$

$p(1) = 3 - 15 + 18 = \underline{6}$

$p(i) = -3i + 15 + 18i = \underline{15 + 15i}$

$p(-1) = -3 - 15 - 18 = \underline{-36}$

$p(-i) = 3i + 15 - 18i = \underline{15 - 15i}$

$$DFT_4(p) = (6, 15+15i, -36, 15-15i)$$

How do we compute DFT efficiently?

Polynomial multiplication

Compute the product of two polynomials *p, q* of degree < *n*:

p, q of degree *n-1*, *n* coefficients

$\downarrow$ **Evaluation:** $\underbrace{\omega_{2n}^{0}, \omega_{2n}^{1}, \ldots, \omega_{2n}^{2n-1}}$

$2n$ point-value pairs $\left(\omega_{2n}^{i}, p(\omega_{2n}^{i})\right)$ and $\left(\omega_{2n}^{i}, q(\omega_{2n}^{i})\right)$

$\downarrow$ **Pointwise multiplication**

$2n$ point-value pairs $\left(\omega_{2n}^{i}, pq(\omega_{2n}^{i})\right)$

$\downarrow$ **Interpolation**

pq of degree $2n$-2, $2n$-1 coefficients

4. Properties of the roots of unity

$\omega_{2n}^{0}, \omega_{2n}^{1}, \ldots, \omega_{2n}^{2n-1}$ form a multiplicative group

Cancellation lemma:
For any integers $n > 0$, $k \geq 0$ and $d > 0$ we have:

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

Proof:

$$\omega_{dn}^{dk} = e^{2\pi i \cancel{d}k/(\cancel{d}n)} = e^{2\pi i k/n} = \omega_{n}^{k}$$

Therefore:

$$\omega_{2n}^{n} = \omega_{2}^{1} = -1$$

Group

(G, *) G set, * Operation $*: G \times G \longrightarrow G$

(1) Closure

$$a, b \in G \Longrightarrow a * b \in G$$

(2) Associativity

$$(a * b) * c = a * (b * c)$$

(3) Neutral Element

$$\exists e \in G : \quad e * a = a * e = a$$

(4) Inverse Element

$$\forall a \in G \quad \exists b \in G : \quad a * b = b * a = e$$

Group of roots of unity

$G = \{ \underline{\omega_{2n}^{0}}, \underline{\omega_{2n}^{1}}, \dots, \underline{\omega_{2n}^{2n-1}} \}$ * Complex multiplication

(1) Closure

$\omega_{2n}^{k} \cdot \omega_{2n}^{\ell} = \omega_{2n}^{k+\ell}$ $\quad k+\ell < 2n$ ✓

$\quad k+\ell \geq 2n$

$= \omega_{2n}^{(k+\ell) \bmod 2n}$

(2) Associativity

$(\omega_{2n}^{k} \cdot \omega_{2n}^{\ell}) \cdot \omega_{2n}^{m} = \omega_{2n}^{(k+\ell) \bmod 2n} \cdot \omega_{2n}^{m} = \omega_{2n}^{(k+\ell+m) \bmod 2n}$

$\omega_{2n}^{k} \cdot (\omega_{2n}^{\ell} \cdot \omega_{2n}^{m}) = \omega_{2n}^{k} \cdot \omega_{2n}^{(\ell+m) \bmod 2n} = \omega_{2n}^{(k+\ell+m) \bmod 2n}$

(3) Neutral Element

$\omega_{2n}^{0} = 1$

(4) Inverse Element

$\omega_{2n}^{k}: \quad 1 = \omega_{2n}^{k-k} = \omega_{2n}^{k} \cdot \omega_{2n}^{-k} = \omega_{2n}^{k} \cdot \omega_{2n}^{2n-k}$

5. Discrete Fourier Transform

$$DFT_n(p) = \left(p(\omega_n^0), p(\omega_n^1), \ldots, p(\omega_n^{n-1})\right)$$

$p(\omega_n^k)$

Fast Fourier Transform:

Computation of *DFT$_n$(p)* by means of a divide-and-conquer approach.

$$DFT_{\frac{n}{2}}(p') = \left(p(\omega_{\frac{n}{2}}^0), \ldots, p(\omega_{\frac{n}{2}}^{\frac{n}{2}-1})\right)$$

Discrete Fourier Transform

Idea: (assume *n* is even)

we will evaluate

$x \in \{\omega_n^0, \ldots, \omega_n^{n-1}\}$

$x = \omega_n^k$

$$\underbrace{p(x)} = a_0 + a_1 x + \ldots + a_{n-1} x^{n-1}$$
$$= a_0 + a_2 x^2 + \ldots + a_{n-2} x^{n-2} +$$
$$a_1 x + a_3 x^3 + \ldots + a_{n-1} x^{n-1}$$
$$= a_0 + a_2 x^2 + \ldots + a_{n-2} \left(x^2\right)^{(n-2)/2} +$$
$$x\left(a_1 + a_3 x^2 + \ldots + a_{n-1} \left(x^2\right)^{(n-2)/2}\right)$$

$(\omega_n^k)^2$

$$= \underbrace{p_0\left(x^2\right)} + \underbrace{x p_1\left(x^2\right)}$$

$$p_0(x) = a_0 + a_2 x + \ldots + a_{n-2} x^{(n-2)/2}$$
$$p_1(x) = a_1 + a_3 x + \ldots + a_{n-1} x^{(n-2)/2}$$

Discrete Fourier Transform

n even

Evaluation for $k = 0, \dots, n-1$:

$$\omega_{n/2}^{0}, \dots, \omega_{n/2}^{n/2-1}$$

$$p(\omega_n^k) = p_0\left((\omega_n^k)^2\right) + \omega_n^k\, p_1\left((\omega_n^k)^2\right) = \begin{cases} p_0(\omega_{n/2}^k) + \omega_n^k\, p_1(\omega_{n/2}^k), & \text{if } k < n/2 \\ p_0(\omega_{n/2}^{k-n/2}) + \omega_n^k\, p_1(\omega_{n/2}^{k-n/2}), & \text{if } k \ge n/2 \end{cases}$$

$$(\omega_n^k)^2 = \omega_n^{2k} = \omega_{n/2}^k$$

$$\omega_{n/2}^{k-n/2} = \omega_{n/2}^k \cdot \omega_{n/2}^{-n/2} = \omega_{n/2}^k \cdot \underbrace{(\omega_{n/2}^{n/2})^{-1}}_{=1} = \omega_{n/2}^k$$

$$DFT_n(p) = (\underbrace{p_0(\omega_{n/2}^0), \dots, p_0(\omega_{n/2}^{n/2-1})}_{DFT_{\frac{n}{2}}(p_0),\ k=0,\dots,\frac{n}{2}-1},\ \underbrace{p_0(\omega_{n/2}^0), \dots, p_0(\omega_{n/2}^{n/2-1})}_{DFT_{\frac{n}{2}}(p_0)})$$

$$+\ (\underbrace{\omega_n^0 p_1(\omega_{n/2}^0), \dots, \omega_n^{n/2-1} \cdot p_1(\omega_{n/2}^{n/2-1})}_{DFT_{\frac{n}{2}}(p_1)},\ \underbrace{\omega_n^{n/2} \cdot p_1(\omega_{n/2}^0), \dots, \omega_n^{n-1} p_1(\omega_{n/2}^{n/2-1})}_{DFT_{\frac{n}{2}}(p_1)})$$

Discrete Fourier Transform

Example: $n = 4$

$$p(\omega_4^0) = p_0(\omega_2^0) + \omega_4^0 p_1(\omega_2^0)$$

$$p(\omega_4^1) = p_0(\omega_2^1) + \omega_4^1 p_1(\omega_2^1)$$

$$p(\omega_4^2) = p_0(\omega_2^0) + \omega_4^2 p_1(\omega_2^0)$$

$$p(\omega_4^3) = p_0(\omega_2^1) + \omega_4^3 p_1(\omega_2^1)$$

Computation of *DFT*$_n$

$$DFT_n(p) = \left(p(\omega_n^0), p(\omega_n^1), \ldots, p(\omega_n^{n-1})\right)$$

Base case: $n = 1$ (degree(p) = $n - 1 = 0$)

$DFT_1(p) = a_0$

General case :

Divide:

Divide p into p_0 and p_1

Conquer:

Recursively compute $DFT_{n/2}(p_0)$ and $DFT_{n/2}(p_1)$.

Merge:

For $k = 0, \ldots, n-1$ compute:

$$\left.\begin{aligned} DFT_n(p)_k = {} & \left(DFT_{n/2}(p_0), DFT_{n/2}(p_0)\right)_k + \\ & \omega_n^k \cdot \left(DFT_{n/2}(p_1), DFT_{n/2}(p_1)\right)_k \end{aligned}\right\} O(n)$$

A further improvement

$$p(\omega_n^k) = \begin{cases} p_0(\omega_{n/2}^k) + \omega_n^k p_1(\omega_{n/2}^k) & \text{if } k < n/2 \\ p_0(\omega_{n/2}^{k-n/2}) + \omega_n^k p_1(\omega_{n/2}^{k-n/2}) & \text{if } k \geq n/2 \end{cases}$$

$$= \begin{cases} p_0(\omega_{n/2}^k) + \omega_n^k p_1(\omega_{n/2}^k) & \text{if } k < n/2 \\ p_0(\omega_{n/2}^{k-n/2}) - \omega_n^{k-n/2} p_1(\omega_{n/2}^{k-n/2}) & \text{if } k \geq n/2 \end{cases}$$

$$-\omega_n^{k-n/2} = -\omega_n^k \cdot \omega_n^{n/2} = -\omega_n^k \cdot \underbrace{\omega_2^1}_{=-1} = \omega_n^k$$

$k = 0, \ldots, \frac{n}{2} - 1$

Thus, if $k < n/2$:

$$\rightarrow \underline{p_0(\omega_{n/2}^k)} + \omega_n^k \underline{p_1(\omega_{n/2}^k)} = p(\omega_n^k)$$

$$\rightarrow p_0(\omega_{n/2}^k) - \omega_n^k p_1(\omega_{n/2}^k) = p(\omega_n^{k+n/2})$$

A further improvement

Example:

$$p(\omega_4^0) = p_0(\omega_2^0) + \omega_4^0 p_1(\omega_2^0)$$

$$p(\omega_4^1) = p_0(\omega_2^1) + \omega_4^1 p_1(\omega_2^1)$$

$$p(\omega_4^2) = p_0(\omega_2^0) - \omega_4^0 p_1(\omega_2^0)$$

$$p(\omega_4^3) = p_0(\omega_2^1) - \omega_4^1 p_1(\omega_2^1)$$

6. Fast Fourier Transform

Algorithm: *FFT*(a, n)

Input: Array *a* containing the *n* coefficients of a polynomial *p* and $n = 2^k$

Output: $DFT_n(p)$

1. **if** $n = 1$ **then** /* *p* is constant */
2. **return** *a*
3. $d^{[0]} = FFT([a_0, a_2, \ldots, a_{n-2}], n/2)$ p_0
4. $d^{[1]} = FFT([a_1, a_3, \ldots, a_{n-1}], n/2)$ p_1
5. $\omega_n = e^{2\pi i/n}$
6. $\omega = 1$
7. **for** $k = 0$ **to** $n/2 - 1$ **do** /* $\omega = \omega_n^k$ */
8. $d_k = d_k^{[0]} + \omega \cdot d_k^{[1]}$
9. $d_{k+n/2} = d_k^{[0]} - \omega \cdot d_k^{[1]}$
10. $\omega = \omega_n \cdot \omega$
11. **return** *d*

FFT: Example

$$p(x) = 3x^3 - 15x^2 + 18x + 0$$

a = [0, 18, -15, 3]

$a^{[0]}$ = [0, -15] $a^{[1]}$ = [18, 3]

FFT([0, -15], 2) = (FFT([0],1) + FFT([-15],1), FFT([0],1) - FFT([-15],1))

= (-15,15)

FFT([18, 3],2) = (FFT([18],1) + FFT([3],1), FFT([18],1) - FFT([3],1))

= (21,15)

k = 0 ; ω = 1

d_0 = -15 + 1 * 21 = 6 d_2 = -15 – 1 * 21 = -36

k = 1 ; ω = i

d_1 = 15 + i*15 d_3 = 15 – i*15

FFT(a, 4) = (6, 15+15i, -36, 15 –15i)

7. Analysis

T(n) = Time required for evaluating a polynomial of degree < n at the points $\underline{\omega_{2n}^{0}, \omega_{2n}^{1}, \ldots, \omega_{2n}^{2n-1}}$.

Conquer   Divide & Merge

T(1) = O(1)

T(n) = $\underline{2\,T(n/2)}$ + O(n)

= $\underline{O(n \log n)}$

Polynomial multiplication

Compute the product of two polynomials *p, q* of degree < *n*:

p,q of degree *n-1*, *n* coefficients

$\downarrow$ **Evaluation via FFT:** $\omega_{2n}^{0}, \omega_{2n}^{1}, \ldots, \omega_{2n}^{2n-1}$ $O(n \log n)$

$2n$ point-value pairs $\left(\omega_{2n}^{i}, p(\omega_{2n}^{i})\right)$ and $\left(\omega_{2n}^{i}, q(\omega_{2n}^{i})\right)$

$\downarrow$ **Pointwise multiplication** $O(n)$

$2n$ point-value pairs $\left(\omega_{2n}^{i}, pq(\omega_{2n}^{i})\right)$

$\downarrow$ **Interpolation** ?

pq of degree $2n$-2, $2n$-1 coefficients