0) Fast Fourier Transform

Problem: Evaluate a polynomial

$$Poly(x) = \sum_{i=0}^{n-1} p_i x^i$$ at n distinct points

n is a power of 2

Given $p[0 \ldots n-1]$ array of doubles $p[i] = p_i$

$X[0 \ldots n-1]$ array of values $X[i] = x_i$

Find $\underline{Poly}[poly(X[0]), poly(X[1]) \ldots poly(X[n-1])]$

Write the n^2 method for computing

array $\underline{Poly[i]} = poly(X[i])$ for all $0 \leq i \leq n-1$

① FFT

Goal: Find an $n \log n$ algorithm

D+C: Identify a way of splitting the polynomial into 2 ½ sized problems

$$Poly(x) = \sum_{i=0}^{n-1} p_i x^i$$

split into odd and even polynomials

$$Poly E(x) = p_0 + p_2 x^2 + p_4 x^4 \ldots p_{n-2}$$

even term poly $\sum_{i=0}^{n/2-1} p_{2i} x^{2i}$ odd term poly $\sum_{i=0}^{n/2-1} p_{2i+1} x^{2i+1}$

$$Poly(x) = \sum_{i=0}^{n/2-1} p_{2i} (x^2)^i + x \sum_{i=0}^{n/2-1} p_{2i-1} (x^2)^i$$

$$= Poly E(x^2) + x \, Poly O(x^2)$$

Two smaller problems of ½ size $n/2$

one with just the even coeficients $Poly E$

— " — odd — " — $Poly O$

(2) start writing the pseduo code

Problem decomposition code for odd/even split

Given $p[0 \ldots n-1]$

Write code to "shuffle" the coeficients in $p[\,]$ such that

$PE[0 \ldots n/2-1]$ contains the <u>even</u> coeficients

$PO[0 \ldots n/2-1]$ contains the odd coeficients

Do this with newly allocated arrays.

3 Deal with the evaluate values $X[0 \ldots n-1]$

A problem instance is n coefficients n values.

think of a subproblem 1/2 size

currently we have 1/2 coeficients

BUT NOT 1/2 valules!

Need to compute subproblems with 1/2 points!

HOW?

Note: subproblems evaluate at x^2

and x and $-x$ are distinct

so if $x_i \quad 0 \le i \le n/2 - 1$ and

$$x_{i+n/2} = -x_i$$

$(x_i)^2 = (x_{i+n/2})^2$ so only need half

$$Poly(x_i) = PolyE(x_i^2) + x_i \, PolyO(x_i^2)$$

$$Poly(x_{i+n/2}) = PolyE(x_i^2) - x_i \, PolyO(x_i^2)$$

Compute $PolyE(x_i^2)$ once - use twice

Compute $PolyO(x_i^2)$ once - use twice

4 Write the FFT Code

Poly[] ← FFT (P[], int n, X[])

// allocate memory

// base case

// shuffle odd / even into PE[ ] and PO[]

// compute x_i^2 from x_i X2[] from X[]

// Call recursion on 2×1/2

// Solution construction Poly[i] =

(5) Use correct values of X

initial $n=8$

$$x_0 \quad x_1 \quad x_2 \quad x_3 \quad -x_0 \quad -x_1 \quad -x_2 \quad -x_3$$

next subproblem

$$x_0^2 \quad x_1^2 \quad x_2^2 \quad x_3^2$$

so $x_0^2 = -x_2^2$, $x_1^2 = -x_3^2$

next $x_0^4 \quad x_1^4$

so $x_0^4 = -x_1^4$

Solution $x_0 = 1$, $x_1 = \omega$

$\omega^8 = 1 \qquad \underline{\omega \neq 1}$

In general

$$\omega = e^{\sqrt{-1}\,\frac{2\pi}{n}} = \cos\frac{2\pi}{n} + \sqrt{-1}\,\sin\frac{2\pi}{n}$$

$$\omega^i = e^{i\sqrt{-1}\,\frac{2\pi}{n}} = \cos\frac{2i\pi}{n} + \sqrt{-1}\,\sin\frac{2i\pi}{n}$$

Write code to precompute $X[i] = \omega^i$