Divide and Conquer Algorithms

CSE 101: Design and Analysis of Algorithms

Lecture 13

CSE 101: Design and analysis of algorithms

- Divide and conquer algorithms
  - Reading: Section 2.6
- Quiz 2 is today, last 40 minutes of class
- Homework 6 will be assigned Nov 13

Polynomials

- Polynomial of one variable of degree $n-1$
- Polynomial form

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

$$A(x) = \sum_{k=0}^{n-1} a_k x^k$$

Polynomials

- Polynomial of one variable of degree $n-1$
- Polynomial form
  $A(x) = a_0 + a_1 x^1 + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$
- Coefficients vector form $\langle a_0, a_1, a_2, \ldots, a_{n-1} \rangle$

Operations on polynomials

- Evaluation
- Addition of two polynomials
- Multiplication of two polynomials

Operations on polynomials, evaluation

- $A(x) = a_0 + a_1 x^1 + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$
- Given a polynomial coefficients vector $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ and a value x_0, compute $A(x_0) = a_0 + a_1 x_0^1 + a_2 x_0^2 + \cdots + a_{n-1} x_0^{n-1}$
- How long does this take?

Operations on polynomials, evaluation

- $A(x) = a_0 + a_1 x^1 + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$
- Given a polynomial coefficients vector $\langle a_0, a_1, \dots, a_{n-1} \rangle$ and a value x_0, compute $A(x_0) = a_0 + a_1 x_0^1 + a_2 x_0^2 + \cdots + a_{n-1} x_0^{n-1}$
- How long does this take? $O(n)$

Operations on polynomials, addition of two polynomials

- $A(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{n-1}x^{n-1}$
- $B(x) = b_0 + b_1x^1 + b_2x^2 + \cdots + b_{n-1}x^{n-1}$
- Given the polynomial coefficients vectors $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ and $\langle b_0, b_1, \ldots, b_{n-1} \rangle$, compute their sum
- How long does this take?

Operations on polynomials, addition of two polynomials

- $A(x) = a_0 + a_1 x^1 + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$
- $B(x) = b_0 + b_1 x^1 + b_2 x^2 + \cdots + b_{n-1} x^{n-1}$
- Given the polynomial coefficients vectors $\langle a_0, a_1, \dots, a_{n-1} \rangle$ and $\langle b_0, b_1, \dots, b_{n-1} \rangle$, compute their sum
- How long does this take? $O(n)$

Operations on polynomials, multiplication of two polynomials

- $A(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{n-1}x^{n-1}$
- $B(x) = b_0 + b_1x^1 + b_2x^2 + \cdots + b_{n-1}x^{n-1}$
- Given the polynomial coefficients vectors $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ and $\langle b_0, b_1, \ldots, b_{n-1} \rangle$, compute their product
- If $C(x) = A(x)B(x) = c_0 + c_1x^1 + c_2x^2 + \cdots + c_{2(n-1)}x^{2(n-1)}$, then

  $c_k = \sum_{j=0}^{k} a_j b_{k-j}$ (for $j > \mathrm{n} - 1$, take a_j and b_j to be zero)
- How long does it take to compute the polynomial coefficients vector $\left\langle c_0, c_1, \ldots, c_{2(n-1)} \right\rangle = \langle a_0b_0, a_0b_1 + a_1b_0, \ldots, a_{n-1}b_{n-1} \rangle$?

Operations on polynomials, multiplication of two polynomials

- $A(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{n-1}x^{n-1}$
- $B(x) = b_0 + b_1x^1 + b_2x^2 + \cdots + b_{n-1}x^{n-1}$
- Given the polynomial coefficients vectors $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ and $\langle b_0, b_1, \ldots, b_{n-1} \rangle$, compute their product
- If $C(x) = A(x)B(x) = c_0 + c_1x^1 + c_2x^2 + \cdots + c_{2(n-1)}x^{2(n-1)}$, then

  $c_k = \sum_{j=0}^{k} a_j b_{k-j}$ (for $j > \mathrm{n} - 1$, take a_j and b_j to be zero)
- How long does it take to compute the polynomial coefficients vector $\left\langle c_0, c_1, \ldots, c_{2(n-1)} \right\rangle = \langle a_0b_0, a_0b_1 + a_1b_0, \ldots, a_{n-1}b_{n-1} \rangle$?
  $O(n^2)$

Operations on polynomials

	Coefficient vector
Evaluation	$O(n)$
Addition	$O(n)$
Multiplication	$O(n^2)$

Polynomials

- Polynomial of one variable of degree $n - 1$
- Polynomial form
  $A(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{n-1}x^{n-1}$
- Coefficients vector form $\langle a_0, a_1, a_2, \ldots, a_{n-1} \rangle$
- Another form?

Polynomials

- Polynomial of one variable of degree $n - 1$
- Polynomial form
  $A(x) = a_0 + a_1 x^1 + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$
- Coefficients vector form $\langle a_0, a_1, a_2, \dots, a_{n-1} \rangle$
- Roots vector form $\langle r_0, r_1, r_2, \dots, r_{n-2} \rangle, c$
  where $A(x) = c(x - r_0)(x - r_1)(x - r_2) \dots (x - r_{n-2})$

Polynomials

- Polynomial of one variable of degree $n-1$
- Polynomial form

  $A(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{n-1}x^{n-1}$
- Coefficients vector form $\langle a_0, a_1, a_2, \ldots, a_{n-1} \rangle$
- Roots vector form $\langle r_0, r_1, r_2, \ldots, r_{n-2} \rangle, c$

  where $A(x) = c(x - r_0)(x - r_1)(x - r_2) \ldots (x - r_{n-2})$

Polynomials

- Coefficients vector form $\langle a_0, a_1, a_2, \dots, a_{n-1} \rangle$
- Roots vector form $\langle r_0, r_1, r_2, \dots, r_{n-2} \rangle, c$
  where $A(x) = c(x - r_0)(x - r_1)(x - r_2) \dots (x - r_{n-2})$
- Converting between coefficients vector form and roots vector form is beyond the scope of this course, so assume
  - Coefficients vector form to roots vector form is $O(\infty)$
  - Roots vector form to coefficients vector form is $O(n)$

Operations on polynomials, evaluation

- $A(x) = c(x - r_0)(x - r_1)(x - r_2) \dots (x - r_{n-2})$
- Given a polynomial roots vector $\langle r_0, r_1, r_2, \dots, r_{n-2} \rangle, c$ and a value x_0, compute $A(x_0) = c(x_0 - r_0)(x_0 - r_1)(x_0 - r_2) \dots (x_0 - r_{n-2})$
- How long does this take?

Operations on polynomials, evaluation

- $A(x) = c(x - r_0)(x - r_1)(x - r_2) \dots (x - r_{n-2})$
- Given a polynomial roots vector $\langle r_0, r_1, r_2, \dots, r_{n-2} \rangle, c$ and a value x_0, compute $A(x_0) = c(x_0 - r_0)(x_0 - r_1)(x_0 - r_2) \dots (x_0 - r_{n-2})$
- How long does this take? $O(n)$

Operations on polynomials, addition of two polynomials

- Given the polynomial roots vectors representing two polynomials
- Addition is beyond the scope of this course, so assume $O(\infty)$

Operations on polynomials, multiplication of two polynomials

- Given the polynomial roots vectors representing two polynomials
- How to multiply them?

Operations on polynomials, multiplication of two polynomials

- Given the polynomial roots vectors representing two polynomials
- Multiplication
  - Multiply c values and concatenate polynomial roots vectors

Operations on polynomials, multiplication of two polynomials

- Given the polynomial roots vectors representing two polynomials
- Multiplication
  - Multiply c values and concatenate polynomial roots vectors
  - How long does this take?

Operations on polynomials, multiplication of two polynomials

- Given the polynomial roots vectors representing two polynomials
- Multiplication
  - Multiply c values and concatenate polynomial roots vectors
  - How long does this take? $O(n)$

Operations on polynomials

	Coefficients vector	Roots vector
Evaluation	$O(n)$	$O(n)$
Addition	$O(n)$	$O(\infty)$
Multiplication	$O(n^2)$	$O(n)$

Polynomials

- Polynomial of one variable of degree $n-1$
- Polynomial form

  $A(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{n-1}x^{n-1}$
- Coefficients vector form $\langle a_0, a_1, a_2, \dots, a_{n-1} \rangle$
- Roots vector form $\langle r_0, r_1, r_2, \dots, r_{n-2} \rangle, c$

  where $A(x) = c(x - r_0)(x - r_1)(x - r_2) \dots (x - r_{n-2})$
- Another?

Polynomials

- Polynomial of one variable of degree $n-1$
- Polynomial form
  $A(x) = a_0 + a_1 x^1 + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$
- Coefficients vector form $\langle a_0, a_1, a_2, \dots, a_{n-1} \rangle$
- Roots vector form $\langle r_0, r_1, r_2, \dots, r_{n-2} \rangle, c$
  where $A(x) = c(x - r_0)(x - r_1)(x - r_2) \dots (x - r_{n-2})$
- Samples

Polynomials

- Polynomial of one variable of degree $n-1$
- Polynomial form
  - $A(x) = a_0 + a_1 x^1 + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$
- Samples
  - Given n points $x_0, x_1, \ldots, x_{n-1}$
  - Samples vector
    $\langle A(x_0), A(x_1), \ldots, A(x_{n-1}) \rangle = \langle y_0, y_1, \ldots, y_{n-1} \rangle$

Operations on polynomials

	Coefficients vector	Roots vector	Samples vector
Evaluation	$O(n)$	$O(n)$	$O(n^2)$
Addition	$O(n)$	$O(\infty)$	$O(n)$
Multiplication	$O(n^2)$	$O(n)$	$O(n)$

Polynomials

- Coefficients vector form $\langle a_0, a_1, a_2, \ldots, a_{n-1} \rangle$
- Samples
  - Given n points $x_0, x_1, \ldots, x_{n-1}$
  - Samples vector

    $\langle A(x_0), A(x_1), \ldots, A(x_{n-1}) \rangle = \langle y_0, y_1, \ldots, y_{n-1} \rangle$
- How to convert from coefficients vector form to samples vector form?

Polynomials

- Coefficients vector form $\langle a_0, a_1, a_2, \ldots, a_{n-1} \rangle$
- Samples
  - Given n points $x_0, x_1, \ldots, x_{n-1}$
  - Samples vector $\langle A(x_0), A(x_1), \ldots, A(x_{n-1}) \rangle = \langle y_0, y_1, \ldots, y_{n-1} \rangle$
- Solve a system of linear equations to convert from coefficients vector form to samples vector form

Coefficients vector to samples vector

- Given coefficients $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ and points $\langle x_0, x_1, \ldots, x_{n-1} \rangle$, solve for samples $\langle y_0, y_1, \ldots, y_{n-1} \rangle$ such that $A(x_k) = y_k$

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

Coefficients vector to samples vector

- Given coefficients $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ and points $\langle x_0, x_1, \ldots, x_{n-1} \rangle$, solve for samples $\langle y_0, y_1, \ldots, y_{n-1} \rangle$ such that $A(x_k) = y_k$

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

Coefficients vector to samples vector

- Use *divide and conquer* to convert coefficients vector to samples vector
- Goal: given coefficients $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ and a set of points X so that we compute $\mathrm{y} = A(x)$ for all $x \in X$
- How to split it up?

Coefficients vector to samples vector

- Use divide and conquer to convert coefficients vector to samples vector
- Goal: given coefficients $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ and a set of points X so that we compute $\mathrm{y} = A(x)$ for all $x \in X$
- How to split it up?
  - multiplyKS algorithm

    $\langle a_0, a_1, \ldots, a_{n/2} \rangle$ and $\langle a_{n/2+1}, \ldots, a_{n-1} \rangle$

Coefficients vector to samples vector

- Use divide and conquer to convert coefficients vector to samples vector
- Goal: given coefficients $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ and a set of points X so that we compute $\mathrm{y} = A(x)$ for all $x \in X$
- How to split it up?
  - multiplyKS algorithm

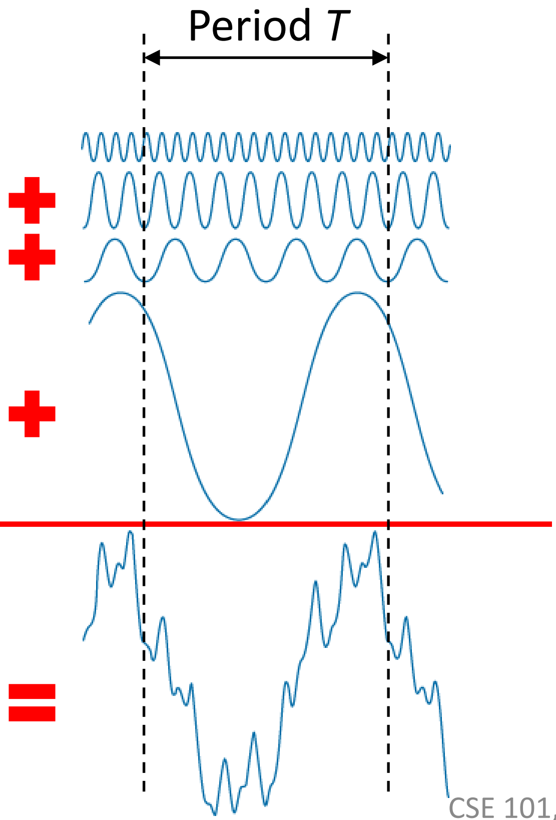
    $\langle a_0, a_1, \ldots, a_{n/2} \rangle$ and $\langle a_{n/2+1}, \ldots, a_{n-1} \rangle$
  - Fast Fourier transform (FFT)

    $\langle a_0, a_2, \ldots, a_{n-2} \rangle$ and $\langle a_1, a_3, \ldots, a_{n-1} \rangle$

Fourier series

Sines and cosines

Periodic function



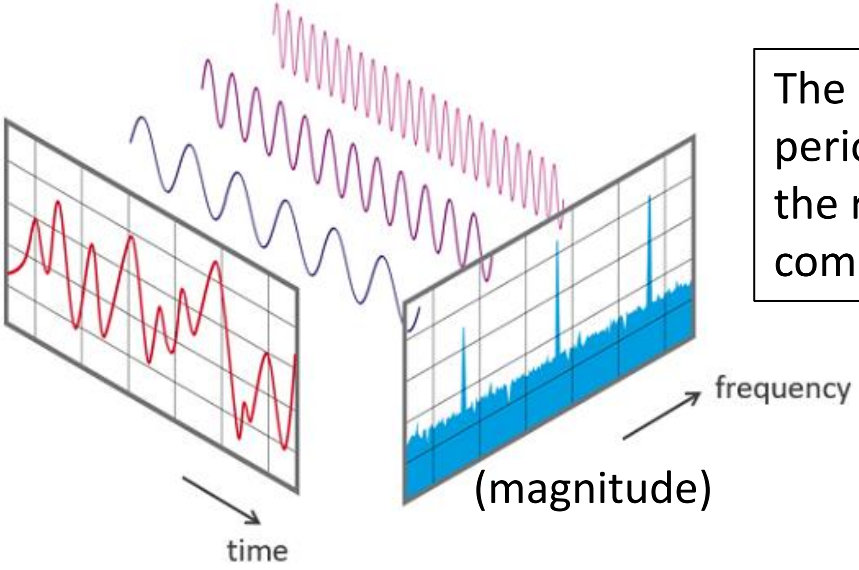
Weighted by magnitude

Shifted by phase

Jean-Baptiste Joseph Fourier
1768-1830

Signals in the frequency domain

Signal in time domain $f(t)$ → Fourier transform → Signal in frequency domain $F(v)$



The Fourier transform reveals periodicities in input data as well as the relative strengths of any periodic components

Signals in the frequency domain

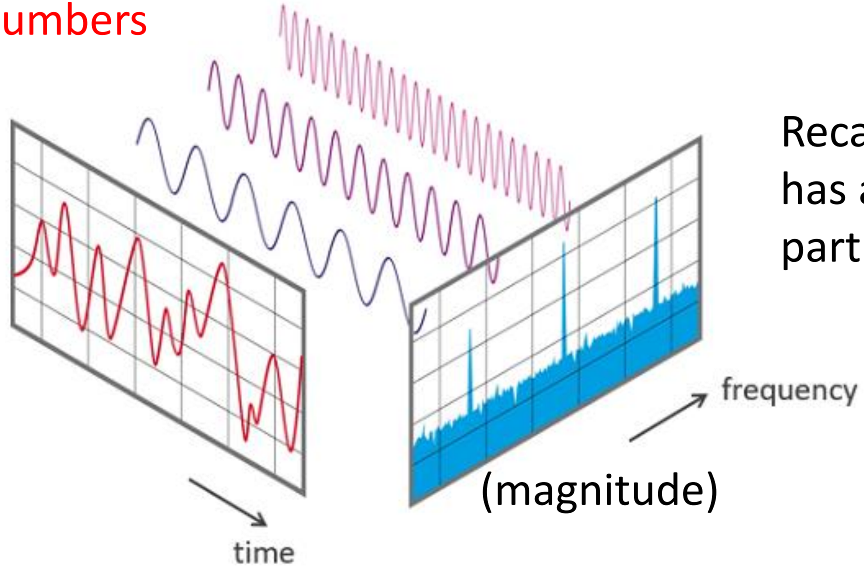
Signal in time domain

$f(t)$ Sequence of real numbers

→ Fourier transform →

Signal in frequency domain

$F(v)$ Sequence of complex numbers



(magnitude)

Recall that a complex number $x + iy$ has a real part x and an imaginary part y

magnitude is $\sqrt{x^2 + y^2}$

phase is $\tan^{-1}\left(\frac{y}{x}\right)$

Signal processing in the frequency domain

Signal in time domain $f(t)$ → Fourier transform → Signal in frequency domain $F(v)$

Signal in frequency domain $F(v)$ → Frequency domain processing → Signal in frequency domain $G(v)$

Signal in frequency domain $G(v)$ → Inverse Fourier transform → Signal in time domain $g(t)$

Fourier transform

- Continuous Fourier transform

$$F(v) = \int_{-\infty}^{\infty} f(t)\, e^{-2\pi i v t} dt$$

- Discrete function

  $f(t) \to f(t_k)$ by letting $f_k = f(t_k)$,
  where $t_k = k\Delta$, with $k = 0, \cdots, N-1$

- Discrete Fourier transform (DFT)

$$F_n = \sum_{k=0}^{N-1} f_k e^{-2\pi i n k / N} \qquad O(n^2)$$

Coefficients vector to samples vector

- Use divide and conquer to convert coefficients vector to samples vector
- Goal: given coefficients $\langle a_0, a_1, \dots, a_{n-1} \rangle$ and a set of points X so that we compute $\mathrm{y} = A(x)$ for all $x \in X$
- How to split it up?
  - Cook-Toom algorithm

    $\langle a_0, a_1, \dots, a_{n/2} \rangle$ and $\langle a_{n/2+1}, \dots, a_{n-1} \rangle$ Halves
  - Fast Fourier transform (FFT)

    $\langle a_0, a_2, \dots, a_{n-2} \rangle$ and $\langle a_1, a_3, \dots, a_{n-1} \rangle$ Evens and odds

Evens and odds

- Goal: given coefficients $\langle a_0, a_1, \dots, a_{n-1} \rangle$ and a set of points X so that we compute $\mathrm{y} = A(x)$ for all $x \in X$
- Split up by evens and odds

  $\langle a_0, a_2, \dots, a_{n-2} \rangle$ and $\langle a_1, a_3, \dots, a_{n-1} \rangle$

  $A_e(x) = a_0 + a_2 x + a_4 x^2 + \cdots$

  $A_o(x) = a_1 + a_3 x + a_5 x^2 + \cdots$
- How to combine $A_e(x), A_o(x)$ to get

  $A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + \cdots$

Evens and odds

- Goal: given coefficients $\langle a_0, a_1, \dots, a_{n-1} \rangle$ and a set of points X so that we compute $\mathrm{y} = A(x)$ for all $x \in X$
- Split up by evens and odds

  $\langle a_0, a_2, \dots, a_{n-2} \rangle$ and $\langle a_1, a_3, \dots, a_{n-1} \rangle$

  $A_e(x) = a_0 + a_2 x + a_4 x^2 + \cdots$ $A_e(x^2) = a_0 + a_2 x^2 + a_4 x^4 + \cdots$

  $A_o(x) = a_1 + a_3 x + a_5 x^2 + \cdots$ $x A_o(x^2) = x a_1 + a_3 x^3 + a_5 x^5 + \cdots$
- How to combine $A_e(x), A_o(x)$ to get

  $A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + \cdots$

Evens and odds

- Goal: given coefficients $\langle a_0, a_1, \dots, a_{n-1} \rangle$ and a set of points X so that we compute $\mathrm{y} = A(x)$ for all $x \in X$
- Split up by evens and odds

  $A_e = \langle a_0, a_2, \dots, a_{n-2} \rangle$ and $\mathrm{A}_o = \langle a_1, a_3, \dots, a_{n-1} \rangle$
- Combine $A_e(x), A_o(x)$

  $A(x) = A_e(x^2) + xA_o(x^2)$
- So, if one could compute $A_e(\alpha)$ and $A_o(\alpha)$ for all $\alpha \in X^2 = \{x^2 | x \in X\}$, then one could combine them to compute $A(x)$ for all $x \in X$

FFT recursion

- Given coefficients $\langle a_0, a_1, \dots, a_{n-1} \rangle$ and a set of points X
  - compute $y = A(x)$ for all $x \in X$
- $A_e = \langle a_0, a_2, \dots, a_{n-2} \rangle$ and $A_o = \langle a_1, a_3, \dots, a_{n-1} \rangle$
- Compute $A_e(\alpha)$ and $A_o(\alpha)$ for all $\alpha \in X^2$
- Set $A(x) = A_e(x^2) + xA_o(x^2)$ for all $x \in X$
- Runtime recurrence?

FFT recursion

- Given coefficients $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ and a set of points X $T(n, |X|)$
  - compute y $= A(x)$ for all $x \in X$
- $A_e = \langle a_0, a_2, \ldots, a_{n-2} \rangle$ and $\mathrm{A}_o = \langle a_1, a_3, \ldots, a_{n-1} \rangle$ $O(n)$
- Compute $A_e(\alpha)$ and $A_o(\alpha)$ for all $\alpha \in X^2$ $2T\left(\frac{n}{2}, |X^2|\right)$
- Set $A(x) = A_e(x^2) + xA_o(x^2)$ for all $x \in X$ $O(|X|)$
- Runtime recurrence?

FFT recursion

- Given coefficients $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ and a set of points X
  - compute $\mathrm{y} = A(x)$ for all $x \in X$
- $A_e = \langle a_0, a_2, \ldots, a_{n-2} \rangle$ and $\mathrm{A}_o = \langle a_1, a_3, \ldots, a_{n-1} \rangle$
- Compute $A_e(\alpha)$ and $A_o(\alpha)$ for all $\alpha \in X^2$
- Set $A(x) = A_e(x^2) + xA_o(x^2)$ for all $x \in X$
- Runtime recurrence?

$T(n, |X|)$

$O(n)$

$2T\left(\frac{n}{2}, |X^2|\right)$

$O(|X|)$

$$T(n, |X|) = 2T\left(\frac{n}{2}, |X^2|\right) + O(n + |X|)$$

FFT recursion

- Given coefficients $\langle a_0, a_1, \dots, a_{n-1} \rangle$ and a set of points X
  - compute $\mathrm{y} = A(x)$ for all $x \in X$
- $A_e = \langle a_0, a_2, \dots, a_{n-2} \rangle$ and $\mathrm{A}_o = \langle a_1, a_3, \dots, a_{n-1} \rangle$
- Compute $A_e(\alpha)$ and $A_o(\alpha)$ for all $\alpha \in X^2$
- Set $A(x) = A_e(x^2) + xA_o(x^2)$ for all $x \in X$
- Runtime recurrence

$$T(n, |X|) = 2T\left(\frac{n}{2}, |X^2|\right) + O(n + |X|)$$

Now, let's put the fast in fast Fourier transform

Collapsing set

- We need a set X with the property that when you square all the elements in the set, you reduce the number of elements until you get down to a set with a single element
- $|X| = 1$ or $|X^2| = \frac{|X|}{2}$ with X^2 also a collapsing set

Collapsing set

- At the leaves of the recursion tree, we want X to have a single element
- Let's say $X = \{1\}$
- Is there a set of elements Z that has the property that $|Z| = 2$ and $Z^2 = X$?

Collapsing set

- At the leaves of the recursion tree, we want X to have a single element
- Let's say $X = \{1\}$
- Is there a set of elements Z that has the property that $|Z| = 2$ and $Z^2 = X$?

$Z = \{-1, 1\}, Z^2 = \{1\}$

Collapsing set

- If $X = \{-1,1\}$ then $X^2 = \{1\}$
- What is the relationship between $\{-1,1\}$ and $\{1\}$?

Collapsing set

- If $X = \{-1,1\}$ then $X^2 = \{1\}$
- What is the relationship between $\{-1,1\}$ and $\{1\}$? -1 and 1 are the two square roots of 1

Collapsing set

- Remember, every nonzero number has two square roots
  - If they are not real, then they are complex
- Going one level further, if $X = \{i, -i, 1, -1\}$, then $X^2 = \{1, -1\}$ and $X^4 = \{1\}$

Collapsing set

$$\{1\}$$

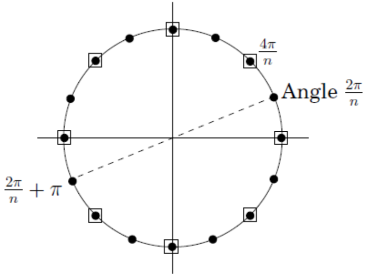
$$\{1, -1\}$$

$$\{1, -1, i, -i\}$$

$$\{1, -1, i, -i, \frac{\sqrt{2}}{2}(1+i), \frac{\sqrt{2}}{2}(-1+i), \frac{\sqrt{2}}{2}(-1-i), \frac{\sqrt{2}}{2}(1-i)\}$$

$$\vdots$$

$$\{\omega_n, \omega_n^2, \dots, \omega_n^n\}$$



Complex plane

The nth complex roots of unity

Solutions to the equation $z^n = 1$.

By the multiplication rule: solutions are $z = (1, \theta)$, for θ a multiple of $2\pi/n$ (shown here for $n = 16$).

For even n:

- These numbers are *plus-minus paired*: $-(1,\theta) = (1, \theta+\pi)$.
- Their squares are the $(n/2)$nd roots of unity, shown here with boxes around them.

FFT recursion

- Given coefficients $\langle a_0, a_1, \dots, a_{n-1}\rangle$ and the n roots of unity $X = \{\omega_n, \omega_n^2, \dots, \omega_n^n\}$, and assume $n = 2^k$
  - compute y $= A(x)$ for all $x \in X$
- $A_e = \langle a_0, a_2, \dots, a_{n-2}\rangle$ and $\mathrm{A}_o = \langle a_1, a_3, \dots, a_{n-1}\rangle$ $O(n)$
- Compute $A_e(\alpha)$ and $A_o(\alpha)$ for all $\alpha \in X^2$ $2T\left(\frac{n}{2}, |X^2|\right)$
- Set $A(x) = A_e(x^2) + xA_o(x^2)$ for all $x \in X$ $O(|X|)$
- Runtime recurrence?

$= \frac{|X|}{2}$

Collapsing set

FFT recursion

$$T(n, |X|) = 2T\left(\frac{n}{2}, |X^2|\right) + O(n + |X|)$$

$$T(n, |X|) = 2T\left(\frac{n}{2}, \frac{|X|}{2}\right) + O(n + |X|)$$

Collapsing set

Now it is fast

FFT recursion

$$T(n, |X|) = 2T\left(\frac{n}{2}, |X^2|\right) + O(n + |X|)$$

$$T(n, |X|) = 2T\left(\frac{n}{2}, \frac{|X|}{2}\right) + O(n + |X|)$$

$|X| = n$

$m = 2n$

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

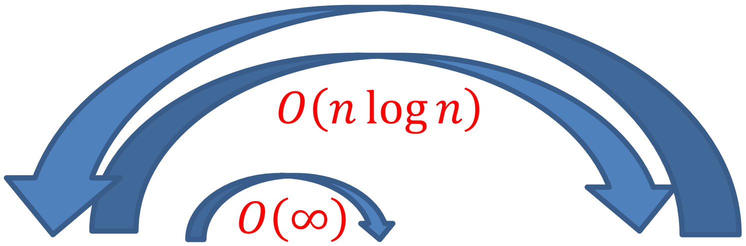
$$T(m) = O(m \log m)$$

$$T(m) = O(n \log n)$$

(using Master theorem)

Converting from the coefficients vector to the samples vector is $O(n \log n)$

Operations on polynomials



	Coefficients vector	Roots vector	Samples vector
Evaluation	$O(n)$	$O(n)$	$O(n^2)$
Addition	$O(n)$	$O(\infty)$	$O(n)$
Multiplication	$O(n^2)$	$O(n)$	$O(n)$

Fast Fourier transform

- The fast Fourier transform (FFT) algorithm reduces the complexity of the discrete Fourier transform (DFT) from $O(n^2)$ to $O(n \log n)$
- FFT is considered one of the most important numerical algorithms of the 20th century

Next lecture

- Divide and conquer algorithms
  - Reading: Sections 2.3 and 2.4