CSE 421
Algorithms

Richard Anderson

Lecture 14

Inversions, Multiplication, FFT

Divide and Conquer Algorithms

- Mergesort, Quicksort
- Strassen's Algorithm
- Closest Pair Algorithm (2d)
- Inversion counting
- Integer Multiplication (Karatsuba's Algorithm)
- FFT
  - Polynomial Multiplication
  - Convolution

Inversion Problem

- Let $a_1, \ldots a_n$ be a permutation of 1 . . n
- (a_i, a_j) is an inversion if $i < j$ and $a_i > a_j$

4, 6, 1, 7, 3, 2, 5

- Problem: given a permutation, count the number of inversions
- This can be done easily in $O(n^2)$ time
  - Can we do better?

Counting Inversions

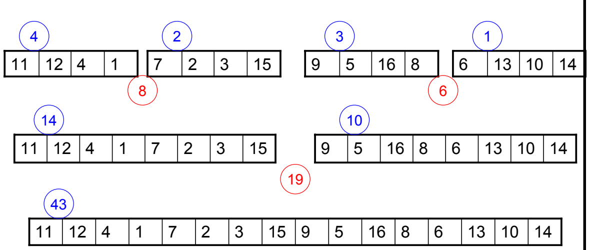
11	12	4	1	7	2	3	15	9	5	16	8	6	13	10	14

Count inversions on lower half

Count inversions on upper half

Count the inversions between the halves

Count the Inversions



Problem – how do we count inversions between sub problems in O(n) time?

- Solution – Count inversions while merging

1	2	3	4	7	11	12	15

5	6	8	9	10	13	14	16

Standard merge algorithm – add to inversion count when an element is moved from the upper array to the solution

Use the merge algorithm to count inversions

1	4	11	12

2	3	7	15

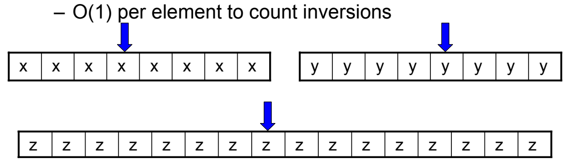
5	8	9	16

6	10	13	14

Indicate the number of inversions for each element detected when merging

Inversions

- Counting inversions between two sorted lists
  - O(1) per element to count inversions



- Algorithm summary
  - Satisfies the "Standard recurrence"
  - T(n) = 2 T(n/2) + cn

Integer Arithmetic

```
    9715480283945084383094856701043643845790217965702956767
+   1242431098234099057329075097179898430928779579277597977
```

Runtime for standard algorithm to add two n digit numbers:

```
    2095067093034680994318596846868779409766717133476767930
X   5920175091777634709677679342929097012308956679993010921
```

Runtime for standard algorithm to multiply two n digit numbers:

Recursive Algorithm (First attempt)

$x = x_1 2^{n/2} + x_0$

$y = y_1 2^{n/2} + y_0$

$xy = (x_1 2^{n/2} + x_0)(y_1 2^{n/2} + y_0)$

$= x_1y_1 2^n + (x_1y_0 + x_0y_1)2^{n/2} + x_0y_0$

Recurrence:

Run time:

Simple algebra

$x = x_1 2^{n/2} + x_0$

$y = y_1 2^{n/2} + y_0$

$xy = x_1y_1 2^n + (x_1y_0 + x_0y_1) 2^{n/2} + x_0y_0$

$p = (x_1 + x_0)(y_1 + y_0) = x_1y_1 + x_1y_0 + x_0y_1 + x_0y_0$

Karatsuba's Algorithm

Multiply n-digit integers x and y

Let $x = x_1 2^{n/2} + x_0$ and $y = y_1 2^{n/2} + y_0$

Recursively compute

$a = x_1y_1$

$b = x_0y_0$

$p = (x_1 + x_0)(y_1 + y_0)$

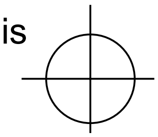
Return $a2^n + (p - a - b)2^{n/2} + b$

Recurrence: T(n) = 3T(n/2) + cn

FFT, Convolution and Polynomial Multiplication

- Preview
  - FFT - O(n log n) algorithm
    - Evaluate a polynomial of degree n at n points in O(n log n) time
  - Computation of Convolution and Polynomial Multiplication (in O(n log n)) time

Complex Analysis



- Polar coordinates: $re^{\theta i}$
- $e^{\theta i} = \cos \theta + i \sin \theta$
- A is a n^{th} root of unity if $a^n = 1$
- Square roots of unity: +1, -1
- Fourth roots of unity: +1, -1, i, -i
- Eighth roots of unity: +1, -1, i, -i, $\beta + i\beta$, $\beta - i\beta$, $-\beta + i\beta$, $-\beta - i\beta$ where β = sqrt(2)

$e^{2\pi ki/n}$

- $e^{2\pi i} = 1$
- $e^{\pi i} = -1$
- n^{th} roots of unity: $e^{2\pi ki/n}$ for k = 0 …n-1
- Notation: $\omega_{k,n} = e^{2\pi ki/n}$
- Interesting fact:

  $1 + \omega_{k,n} + \omega^2_{k,n} + \omega^3_{k,n} + \ldots + \omega^{n-1}_{k,n} = 0$
  for k != 0

Convolution

- $a_0, a_1, a_2, \ldots, a_{m-1}$
- $b_0, b_1, b_2, \ldots, b_{n-1}$
- $c_0, c_1, c_2, \ldots, c_{m+n-2}$ where $c_k = \sum_{i+j=k} a_i b_j$

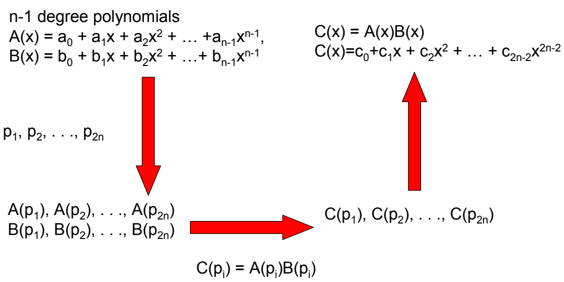
Applications of Convolution

- Polynomial Multiplication
- Signal processing
  - Gaussian smoothing
  - Sequence $a_1, a_2, \ldots, a_n$
  - Mask, $w_{-k}, w_{-(k-1)}, \ldots, w_{-1}, w_0, w_1, \ldots, w_{k-1}, w_k$
- Addition of random variables

FFT Overview

- Polynomial interpolation
  - Given n+1 points (x_i, y_i), there is a unique polynomial P of degree at most n which satisfies $P(x_i) = y_i$

Polynomial Multiplication



FFT

- Polynomial $A(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1}$
- Compute $A(\omega_{j,n})$ for $j = 0, \dots, n-1$