CSE 421
Algorithms

Richard Anderson

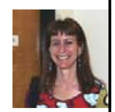
Lecture 14

Divide and Conquer

Announcements

- Mon. February 9
  - Midterm
- Wed. February 11
  - Punya Biswal
  - Divide and Conquer Algorithms
  - Read 5.3 – 5.5
- Fri. February 13
  - Anna Karlin
  - FFT – Read 5.6

Punya Biswal



Midterm exam

- Instructions
  - Closed book, closed notes, no calculators
  - Time limit: 50 minutes
  - Answer the problems on the exam paper
  - If you need extra space use the back of the page
  - Problems are not of equal difficulty, if you get stuck on a problem, move on.
- Seven problems
  - Uniform coverage
  - Several “true/false/justify”
  - Two algorithm design questions

Where I will be . . .

- Digital StudyHall Project
- Lucknow, India

Talk: Richard Anderson, CIS Lecture Series, Wednesday, February 18, 3pm, MGH 420

What you really need to know about recurrences

- Work per level changes geometrically with the level
- Geometrically increasing ($x > 1$)
  - The bottom level wins
- Geometrically decreasing ($x < 1$)
  - The top level wins
- Balanced ($x = 1$)
  - Equal contribution

$T(n) = aT(n/b) + n^c$

- Balanced: $a = b^c$
- Increasing: $a > b^c$
- Decreasing: $a < b^c$

Divide and Conquer Algorithms

- Split into sub problems
- Recursively solve the problem
- Combine solutions

- Make progress in the split and combine stages
  - Quicksort – progress made at the split step
  - Mergesort – progress made at the combine step
- D&C Algorithms
  - Strassen's Algorithm – Matrix Multiplication
  - Inversions
  - Median
  - Closest Pair
  - Integer Multiplication
  - FFT

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?

Application

- Counting inversions can be use to measure how close ranked preferences are
  - People rank 20 movies, based on their rankings you cluster people who like that same type of movie

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

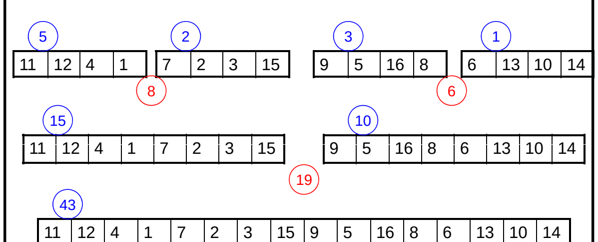
11	12	4	1	7	2	3	15	9	5	16	8	6	13	10	14

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

x	x	x	x	x	x	x	x

y	y	y	y	y	y	y	y

z	z	z	z	z	z	z	z	z	z	z	z	z	z	z	z

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

Computing the Median

- Given n numbers, find the number of rank n/2
- Selection, given n numbers and an integer k, find the k-th largest

Select(A, k)

```
Select(A, k){
        Choose element x from A
        S1 = {y in A | y < x}
        S2 = {y in A | y > x}
        S3 = {y in A | y = x}
        if (|S2| >= k)
                return Select(S2, k)
        else if (|S2| + |S3| >= k)
                return x
        else
                return Select(S1, k - |S2| - |S3|)
}
```

S_1	S_3	S_2

Randomized Selection

- Choose the element at random
- Analysis can show that the algorithm has expected run time O(n)

Deterministic Selection

- What is the run time of select if we can guarantee that choose finds an x such that $|S_1| < 3n/4$ and $|S_2| < 3n/4$

BFPRT Algorithm





- A very clever choose algorithm . . .

Split into n/5 sets of size 5
M be the set of medians of these sets
Let x be the median of M

BFPRT runtime

$|S_1| < 3n/4$, $|S_2| < 3n/4$

Split into n/5 sets of size 5
M be the set of medians of these sets
x be the median of M
Construct S_1 and S_2
Recursive call in S_1 or S_2

BFPRT Recurrence

- $T(n) \le T(3n/4) + T(n/5) + c\,n$

Prove that $T(n) \le 20\,c\,n$