CSE 421
Algorithms
Richard Anderson
Lecture 13
Divide and Conquer

Announcements
• Guest Lecturers
  – Anna Karlin (10/31, 11/2)
  – Venkat Guruswami (10/28, 11/4)
• Homework 5 and Homework 6 are available
• I’m going to try to be clear when student submissions are expected

Instructor Example
Student Submission

Example

What is the solution to: $\sum_{i=0}^{n} x^i$?

What are the asymptotic bounds for $x < 1$ and $x > 1$?

Solve by unrolling
$T(n) = n + 3T(n/4)$

Solve by unrolling
$T(n) = n + 5T(n/2)$

A non-linear additive term
$T(n) = n^2 + 3T(n/2)$
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

Classify the following recurrences (Increasing, Decreasing, Balanced)

- $T(n) = n + 5T(n/8)$
- $T(n) = n + 9T(n/8)$
- $T(n) = n^2 + 4T(n/2)$
- $T(n) = n^3 + 7T(n/2)$
- $T(n) = n^{1/2} + 3T(n/4)$

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

Closest Pair Problem

- Given a set of points find the pair of points $p, q$ that minimizes $\text{dist}(p, q)$

Divide and conquer

- If we solve the problem on two subsets, does it help? (Separate by median x coordinate)

Packing Lemma

Suppose that the minimum distance between points is at least $\delta$, what is the maximum number of points that can be packed in a ball of radius $\delta$?
Combining Solutions

• Suppose the minimum separation from the sub problems is $\delta$
• In looking for cross set closest pairs, we only need to consider points with $\delta$ of the boundary
• How many cross border interactions do we need to test?

Details

• Preprocessing: sort points by $y$
• Merge step
  – Select points in boundary zone
  – For each point in the boundary
    • Find highest point on the other side that is at most $\delta$ above
    • Find lowest point on the other side that is at most $\delta$ below
    • Compare with the points in this interval (there are at most 6)

Algorithm run time

• After preprocessing:
  – $T(n) = cn + 2T(n/2)$

Counting Inversions

Count inversions on lower half
Count inversions on upper half
Count the inversions between the halves
Count the Inversions

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

Problem – how do we count inversions between sub problems in O(n) time?
• Solution – Count inversions while merging

Use the merge algorithm to count inversions

1 4 11 12 2 3 7 15

5 8 9 16 6 10 13 14

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