CS 420: Advanced Algorithm Design and Analysis
Spring 2015 – Lecture 1

Department of Computer Science
University of British Columbia

January 6, 2015

Announcements

Assignments...
Upcoming Exams / Review Sessions ...
Readings...

- check the CS 420 homepage:
  http://people.cs.ubc.ca/~kirk/cs420/
- go carefully through the General Information Handout
- review CS320 notes, particularly material on sorting and searching and basic graph algorithms, including basic data structures and algorithm design strategies: know where you can find what you may need to revisit
- read the essay:
  The Algorithm: Idiom of Modern Science, by Bernard Chazelle
  http://www.cs.princeton.edu/~chazelle/pubs/algorithm.html
  for thoughtful (and amusing) motivation for studying algorithms.

Today...

Administration

- quick overview of course
- highlights of General Information Handout

Start a case study

- finding extrema of a set (and related problems/issues)
- review of basic concepts; preview of others

General Information highlights

- contact info & office hours
- prerequisite info & course objectives
- texts and other reference sources
  - Kleinberg & Tardos
  - Cormen et al (CLRS)
  - Erickson'sNotes
- Lectures (including tentative lecture outline)
- Assignments (eight in total); remarks on collaboration & plagiarism
- Exams (three term exams, in evenings, but NO final)
- Tentative grading scheme

A Case Study: Finding the extrema (outliers) of a set of n elements (and related issues)

- find the maximum of n numbers
  - (several) algorithms solve this problem efficiently:
    - iterative, recursive, tournament...
    - they all use $n-1$ comparisons, in the worst, average and even best cases
  - can we do better?
    - No...there is a *lower bound* of $n-1$ comparisons: need to identify all $n-1$ non-maximums
    - Maybe...depending on the *cost measure*, and the *computation model*

A Case Study: Finding the extrema (outliers) of a set of n elements (and related issues)

- find the minimum
  - exactly the same!
    (by reduction to maximum)

Case study cont.

What about other extrema (or ranking)-type questions?

- find both the maximum *and* the minimum
- find the largest *and* second largest
- find the first, second and third largest
- find the median

Solvable by reduction...

- ... but this may not be optimal
- a more detailed analysis (comparison cost) of such elementary problems shows...
  - find both the maximum *and* the minimum $\lceil 3n/2 \rceil - 2$
  - find the largest *and* second largest $n + \lceil \lg n \rceil - 2$
  - find the first, second and third largest ???
  - find the median worst case is between $2n$ and $3n$; expected case (using a randomized algorithm) is at most $1.5n$

Next class...

Continue case study on finding extrema (reviewing basic issues & previewing others)

- taking the cost of other operations/resources into account
  - *auxiliary space* in finding the max and second largest; *streaming algorithms*; *time-space tradeoffs*
  - *update costs* in finding the maximum (the iterative and on-line hiring problems); *randomized algorithms*
- finding extrema in other computation models
  - parallel algorithms
  - distributed algorithms; communication complexity
- finding extrema in other input domains
  - inputs are drawn from $\mathcal{U} = \{0, 1, \ldots m-1\}$
  - inputs are specified *implicitly*; *linear programming*
  - inputs are points in two (or higher) dimensions; *computational geometry*