CSE332: Data Abstractions

Lecture 28: Course Wrap-Up / Victory Lap

Dan Grossman
Spring 2010

Final Exam

As also indicated in class-list email and on the web page:

- Next Tuesday, 2:30-4:20
- Intention is to test a subset of the topics in sorting, graphs, parallelism, concurrency, amortization
  - In other words, "stuff not covered by the midterm"
  - But as always the course topics build on earlier ones
- You will need to read and write Java, among other things

- Dan leaves town tomorrow morning, but will have email
  - Exam and course grades will not be available until June 13-14
  - Homework and project grades available sooner

Victory Lap

A victory lap is an extra trip around the track
- By the exhausted victors (that's us) ☺

Review course goals
- Slides from Lecture 1
- What I told the faculty

Feedback from you on a new course being taught every quarter
- Anything to discuss as a group
- Course evaluations
  - Please spend even more time than usual on them

Thank you!

Huge thank-you to your TAs
- New homeworks, projects, material, libraries
- Tyler: Use section to teach relevant programming idioms, project 3 guinea pig, teaching the course in summer, …
- Brent: A cool GUI, project 2 unit-test examples, …

Thank you!

And huge thank you to all of **you**

- Great attitude about a new course
- Good class attendance and questions
- Occasionally laughed at stuff ☺

Four slides from Lecture 1

We have 10 weeks to learn *fundamental data structures and algorithms for organizing and processing information*

- "Classic" data structures / algorithms and how to analyze rigorously their efficiency and when to use them
- Queues, dictionaries, graphs, sorting, etc.
- Parallelism and concurrency (new!)

Four slides from Lecture 1

- Introduction to many (not all) of the basic data structures used in computer software
  - Understand the data structures and the trade-offs they make
  - Rigorously analyze the algorithms that use them (math!)
  - Learn how to pick "the right thing for the job"
  - More thorough and rigorous take on topics introduced in 143
    - And more
- Practice design and analysis of data structures / algorithms
- Practice implementing and using these data structures by writing programs
- Experience the purposes and headaches of multithreading

Four slides from Lecture 1

- To be able to make good design choices as a developer, project manager, etc.
  - Reason in terms of the general abstractions that come up in all non-trivial software (and many non-software) systems
- To be able to justify and communicate your design decisions

Dan's take:

3 years from now this course will seem like it was a waste of your time because you can't imagine not "just knowing" every main concept in it

- Key abstractions computer scientists and engineers use almost every day
- A big piece of what separates us from others

Four slides from Lecture 1

(Often highly *non-obvious*) ways to organize information in order to enable *efficient* computation over that information

- Key goal over the next week is introducing asymptotic analysis to *precisely* and *generally* describe efficient use of time and space

A data structure supports certain *operations*, each with a:

- Meaning: what does the operation do/return
- Performance: how efficient is the operation

Victory Lap

A victory lap is an extra trip around the track

- By the exhausted victors (that's us) ☺

Review course goals

- Slides from Lecture 1
- What I told the faculty

Feedback from you on a new course being taught every quarter

- Anything to discuss as a group
- Course evaluations
  - Please spend even more time than usual on them

What I told the faculty about CSE332

Catalog Description:

> **Abstract data types and structures** including dictionaries, balanced trees, hash tables, priority queues, and graphs; sorting; asymptotic analysis; fundamental graph algorithms including graph search, shortest path, and minimum spanning trees; **concurrency and synchronization; and parallelism**.

Goals:

- Deep understanding of core data-structure trade-offs
- Fluency with asymptotic complexity, exponentials, etc.
- Ability to analyze correctness (?) and efficiency
- Recognizing basic opportunities for parallelism
- Addressing challenges of concurrent access to resources

Topics: Data structures + Threads

326 & 332 (20 lectures)

Big-Oh, Algorithm Analysis
Binary Heaps (Priority Qs)
AVL Trees
B Trees
Hashing
Sorting
Graph Traversals
Topological Sort
Shortest Paths
Minimum Spanning Trees
Amortization

Topics: Data structures + Threads

326 & 332 (20 lectures)	Removed from 326 (8 lectures)
Big-Oh, Algorithm Analysis	D-heaps
Binary Heaps (Priority Qs)	Leftist heaps
AVL Trees	Skew heaps
B Trees	Binomial queues
Hashing	Splay trees
Sorting	Disjoint sets
Graph Traversals	Network flow
Topological Sort	Hack job on NP (moves to CSE312)
Shortest Paths	
Minimum Spanning Trees	
Amortization	

Topics: Data structures + Threads

326 & 332 (20 lectures)	Added to 332 (8 lectures)
Big-Oh, Algorithm Analysis	Multithreading Basics (1)
Binary Heaps (Priority Qs)	Fork-Join Parallelism (3)
AVL Trees	• Using Java library
B Trees	• Analysis: **T1** and **T∞**
Hashing	• Amdahl's Law
Sorting	• Reductions, Prefix, Sorting
Graph Traversals	Concurrency (4)
Topological Sort	• Races, deadlocks
Shortest Paths	• Locks (mostly)
Minimum Spanning Trees	• Condition variables (a bit)
Amortization	• Programming guidelines (!)

What works

- Triage 30% without killing the patient
  - Plus pretty deep scrub of other 70%
  - Projects use more modern Java (generics, iterators, JUnit)
  - Still keystone course with algorithms, code, proofs, and charts
- Parallelism and concurrency in this course
  - Natural fit (!): same notion of trade-offs, asymptotics
    - Example: Sequential cut-off
    - Example: Bounded buffer for condition variables
    - Example: Amdahl's Law in the limit
  - Data structures & algorithms are canonical examples
    - Divide-and-conquer, atomic operations, etc.
  - All at the Java / pseudocode level

What might not work

- No textbook for the parallelism and concurrency (no complaints?)
- Aimed for "teachable by others," but never quite sure
- 1.5 / 3 new projects and lots of new slides
  - No disasters, but could use some sanding
- Due to project scheduling, graphs in weeks 6 and 10
- Did not fit: map/reduce, declarative queries
  - Leave to (optional) CSE344

Victory Lap

A victory lap is an extra trip around the track

- By the exhausted victors (that's us) ☺

Review course goals

- Slides from Lecture 1
- What I told the faculty

Feedback from you on a new course being taught every quarter

- Anything to discuss as a group
- Course evaluations
  - Please spend even more time than usual on them

Last slide

What do you think was good about 332?

What could be improved?

And:

Don't be a stranger: let me know how the rest of your time in CSE (and beyond!) goes… I really do like to know.