CS61B Lecture 40

Friday, May 1, 2020

Announcements

- Look out for opportunities to sign up as course staff in the Fall! Tutors and lab assistants are needed.

Summary

What have we done in CS61B?

- Programming language: Java
- Program analysis
- Categories of data structure: Java library structure
- Sequences
- Trees
- Searching
- Sorting
- Pseudo-random numbers
- Graphs
- Pragmatic implementation topics, or practical experience you have gained from writing projects

Java topics

- Object-based programming: organizing around data types
- Object-oriented programming:
    - Dynamic vs. static type
    - Inheritance
    - Idea of interface vs. implementation
- Generic programming (the `<...>` stuff)
- Memory model: containers, pointers, arrays
- Numeric types
- Java syntax and semnatics
- Scope and extent
- Standard idioms, patterns:
    - Objects used as functions (`Comparator`)
    - Partial implementations (`AbstractList`)
    - Iterators
    - Views (e.g. sublists)

Analysis and Algorithmic Techniques

- Asymptotic analysis
- Big Theta, Big O, Big Omega and Little O notations

- Worst case, average case
- Amortized time
- Memoization and dynamic programming

Major Categories of Data Structure

- Collection interface and its subtypes
- Map interface and its subtypes
- Generic skeleton implementations of collections, lists, maps (e.g. `AbstractList`)
- Complete concrete collection and map classes in Java library

Sequences

- Linking
    - Single and double link manipulations
    - Sentinels
- Linking vs. Arrays
- Stacks, queues, deques
- Cricular buffering
- Trade-offs: costs of basic operations

Trees

- Uses of trees: search, representing hierarchical structures.
- Basic operations: insertion and deletion
- Tree traversals
- Representing trees
- Game trees

Searching

- Search trees, range searching
- Multidimensional searches: quad trees
- Hashing
- Priority Queues and Heaps
- Balanced trees
    - Rebalancing by rotation (red-black trees)
    - Balance by construction (B-trees)
    - Probabilistic balance (skip lists)
    - Tries
- Search times and trade-offs

Sorting

- Uses of sorting
- Insertion sort
- Selection sort
- Merge sort
- Heap sort
- Quicksort and selection
- Distribution sort

- Radix sort
- Complexity of various algorithms and when to use them

Random numbers

- Possible uses
- Idea of a pseudo-random sequence
- Linear congruential and additive generators
- Changing distributions:
    - Changing the range
    - Non-uniform distributions
- Shuffling, random selection

Graph structures

- Definition
- Uses: things represented by graphs
- Graph traversal: generic traversal template
- Depth-first traversal, breadth-first traversal
- Topological sort
- Shortest paths
- Minimum spanning trees, union-find structures
- Memory management as a graph problem.

Debugging

- What they do
- How to use to pin down bugs
- Details of some debuggers (Eclipse, IntelliJ etc.)
- Unit testing and JUnit mechanics

Version Control

- What it's for
- Basic concepts behind our particular system
    - Working copy vs. repository copy
    - Committing changes
    - Updating and merging changes
    - Tagging

As a Case Study

We presented the Git version control system as ane xample of a design using several ideas from this course.

- **Graph** (DAG) and **tree** structures represented with files as vertices and strings (file names), rather than machine addresses, as pointers.
- Use of hashing to create a very, very likely to be unique set of names: **probabilistic data structure**.
- Compression uses various kinds of map to facilitate conversion to and from the compressed form, including arrays, tries and hash tables.
- Priority queue in Huffman coding

Assorted Side Trips

- Compression
- Parallel processing
- Storage management and garbage collection

What's after CS61B?

Upper division has several courses being offered in Fall 2020:

- CS C100: Principles and Techniques of Data Science
- CS 160: User Interface Design
- CS 161: Computer Security
- CS 162: OS and System Programming
- CS 164: Programming Languages and Compilers (taught by Prof. Hilfinger!)
- CS 170: Algorithms
- CS 176: Computational Biology
- CS 186: Databases
- CS 188: AI
- CS 169A: Software Engineering
- CS 189: Machine Learning
- CS 194: Assorted Special Topics
- CS 195: Social Implications of Computer Technology
- Numerous graduate courses,
- And EE courses!

And there are several courses not offered in the Fall:

- CS 168: Introduction to the Internet
- CS 152: Computer Architecture
- CS 171: Cryptography
- CS 172: Computability and Complexity
- CS 174: Combinatorics and Discrete Probability
- CS W182: Deep Neural Networks
- CS 184: Graphics
- CS C191: Quantum Information Science

And don't forget, EE and CS are just two of over 150 subjects! You should also involve yourself with internships, to get specific skills and exposure to real problems.