

Lecture #43: Course Summary

- **Extra points** for filling out on-line survey (see announcements on our home page for the link).
- **Course Review:** Friday, 14 May 2007, 1-3PM in 306 Soda.
- **Study suggestion:** The auto-graders for the homeworks are still running. Do them for real!
- **Tournament:** Turn in your tournament version (4 minutes per player per game time limit) using `new-submit` tournament. Due date: Thursday, 13 December by midnight. Results by the final (I hope).
- **Readers and lab assistants needed.** Consider volunteering to be a reader or lab assistant for CS 3, CS 4, CS 61A, or CS 61B next semester. Reader applications will be available at the beginning of the semester (in fact, before). Readers are paid; lab assistants can get unit credit.
- **Programming Contest:** Visit my web page for information about the annual programming contest, which we hold each fall. There are large collections of programming problems you can try your hand on.

Course Topic Summary

- Programming language: Java
- Program Analysis
- Categories of data structure: Java library structure
- Sequences
- Trees
- Searching
- Sorting
- Pseudo-random numbers
- Threads
- Graphs
- Pragmatic implementation topics

Programming-Language 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 semantics
- Scope and extent
- Standard idioms, patterns:
 - Objects used as functions (e.g., Comparator)
 - Partial implementations (e.g., AbstractList)
 - Iterators
 - Views (e.g., sublists)

Analysis

- Asymptotic analysis
- $O(\cdot)$, $o(\cdot)$, $\Omega(\cdot)$, $\Theta(\cdot)$ notations
- Worst case, average case.
- Amortized time

Major Categories of Data Structure

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

Sequences

- Linking:
 - Single and double link manipulations
 - Sentinels
- Linking vs. arrays
- Stacks, queues, dequeues
- Circular buffering
- Trade-offs: costs of basic operations

Trees

- Uses of trees: search, representing hierarchical structures
- Basic operations: insertion, deletion
- Tree traversals
- Representing 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, trade-offs

Sorting

- Uses of sorting
- Insertion sort
- Selection sorting
- Merge sort
- Heap sort
- Quicksort and selection
- Distribution sort
- Radix sort
- Complexity of various algorithms, 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

Threading

- Creating multiple threads of control in Java
- Need and mechanisms for mutual exclusion in Java
- Use of mailboxes for communication

Graph structures

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

Debugging

- What debuggers can do
- How to use to pin down bugs
- Details of some debugger (Eclipse, gjdb, various Windows/Sun products).
- Unit testing: what it means, how to use it.
- JUnit mechanics.

Version Control

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