

Lecture #40: Course Summary

language: Java
analysis
of data structure: Java library structure

from numbers
implementation topics

Analysis and Algorithmic Techniques

analysis
, $\Theta(\cdot)$ notations
average case.
time
and dynamic programming.

Sequences

of double link manipulations
arrays
lists, deques
inserting
costs of basic operations

Trees

uses: search, representing hierarchical structures
operations: insertion, deletion
traversal
of trees

Announcements

on December 5, 6, and 7 will be organized as follows:
I will work on an exam-like set of exercises covering linked
lists, queues, binary trees, binary search trees. Solutions
will be thoroughly reviewed. 1 bonus point (out of 200) for complet-
ing exercises.
Please report any bug for problems with submission, your code, the
use of any of our software.
lab assistants needed. Consider volunteering to be
a lab assistant for CS 10, self-paced courses, CS 61A, or
this semester.
Contest: Visit my web page for information about the
programming contest, which we hold each fall. There are
hundreds of programming problems you can try your hand on.

Programming-Language Topics

of programming: organizing around data types
structured programming:
static type
system
interface vs. implementation
programming (the `<...>` stuff).
examples: containers, pointers, arrays
expressions
syntax and semantics
abstraction
abstractions, patterns:
expressed as functions (e.g., Comparator)
implementations (e.g., AbstractList)
(e.g., sublists)

Major Categories of Data Structure

interface and its subtypes
queue and its subtypes
concrete implementations of collections, lists, maps (AbstractList,
AbstractMap, etc.)
concrete collection and map classes in Java library

Sorting

ing
rt
rting

nd selection
sort

f various algorithms, when to use them?

Graph structures

represented by graphs
sal: the generic traversal template
traversal, breadth-first traversal
ort
ths
ning trees, union-find structures
agement as a graph problem.

Version Control

?
ts behind our particular system:
copy vs. repository copy
g changes
and merging changes.

Searching

s, range searching
onal searches: quad trees.

es and heaps
es
ng by rotation (red-black trees)
y construction (B-trees)
tic balance (skip lists)

s, trade-offs

Random numbers

s
eudo-random sequence
uential and additive generators
tributions:
the range
rm distributions
ndom selection

Debugging

gers can do
o pin down bugs
me debugger (Eclipse, gjdb, various Windows/Sun prod-

what it means, how to use it.
nics.

Assorted Side Trips

essing.
agement and garbage collection.

What's After the Lower Division? (II)

puter Architecture (Asanovic)

graduate courses: including advanced versions of 152,
, 184, 186, 189; plus Cryptography, VLSI design and
topics.

se, EE courses!

rtunities for participating in research and independent

A Case Study

t version-control system as an example of a design using
s from this course.

and **tree** structures represented with files as vertices
file names), rather than machine addresses, as pointers.

ing to create unique (or very, very likely to be unique)
abilistic data structure.

uses various kinds of **map** to facilitate conversion to
npressed form, including **arrays**, **tries**, and **hash tables**
e in Huffman coding.

What's After the Lower Division?

Interface Design (Hartmann)

uter Security (Popa)

ating Systems and System Programming (Joseph, Ragan-

ramming Languages and Compilers (Hilfinger)

icient Algorithms and Intractable Problems (Chiesa, Vazi-

inatorics and Discrete Probability (Friedman)

hics (Ng)

bases

ficial Intelligence (Dragan, Levine)

ine Learning

orted Special Topics: Computational Design and Fabri-
cating, Visualizing and Understanding Deep Neural Net-

What's After the Lower Division? (III)

CS are just two of over 150 subjects!

offer more specific skills and exposure to real prob-

think that CS is a creative activity that (to the true
t to fun!