Lecture #40: Course Summary

language: Java
lysis
f data structure: Java library structure

om numbers

plementation topics

Announcements

filling out our CS61B survey. Watch the website.

alysis and Algorithmic Techniques

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

Programming-Language Topics

d programming: organizing around data types
nted programming:
s. static type
ce
terface vs. implementation
ramming (the <···> stuff).
el: containers, pointers, arrays
es
and semantics
ktent
oms, patterns:
sed as functions (e.g., Comparator)
plementations (e.g., AbstractList)

., sublists)

Sequences

double link manipulations

rrays
es, deques
fering
costs of basic operations

Trees

s: search, representing hierarchical structures
ions: insertion, deletion
sals
trees

ajor Categories of Data Structure

terface and its subtypes
e and its subtypes
eton implementations of collections, lists, maps (`AbstractList`,

ncrete collection and map classes in Java library

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

Sorting

ing
rt
rting

d selection
sort

f various algorithms, when to use them?

Random numbers

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

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.

Debugging

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

what it means, how to use it.
nics.

Version Control

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

Assorted Side Trips

essing.

agement and garbage collection.

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.

at's After the Lower Division? (II)

cient Algorithms and Intractable Problems (Raghavendra)

tography

putability and Complexity (Tal)

binatorics and Discrete Probability (Song)

putational Biology (Yosef)

Neural Networks (Zhang)

dations of Computer Graphics (Ng)

bases (Jain)

ficial Intelligence (Russell, Song)

hine Learning (Shewchuk, Zhang)

ntum Information Science and Technology

What's After the Lower Division?

a C100: Principles & Techniques of Data Science (Perez,

puter Architecture (Wawrzynek)

Interface Design (Canny)

uter Security (Weaver)

rating Systems and System Programming (Joseph, Kubiatowicz)

ramming Languages and Compilers (Chasins)

. to the Internet: Architecture and Protocols

169L: Software Engineering (Ball, A. Fox, P. Fox)

And Beyond!

S are just two of over 150 subjects!

offer more specific skills and exposure to real problems.

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

t's After the Lower Division? (III)

rted Special Topics: Computer Vision and Computational
(Efros), Parallel Programming (Yelick)

l Implications of Computer Technology (Hug)

raduate courses: including advanced versions of 152,
0, 184, 186, 189; plus Cryptography, VLSI design and
topics.

se, EE courses!

rtunities for participating in research and independent
r directed group studies (198).