Department of Industrial and Systems Engineering
Spring 2007

Algorithms in Systems Engineering

(IE 170)

Meeting:	Monday, Wednesday, Friday 11:10AM-12PM	375 Packard Lab
	Monday 1-4PM	444 Mohler Lab

Jeff Linderoth
Office: 325 Mohler
Office hours: Wednesday and Friday 10–11AM, (also by appt.)
Phone: 610-758-4879
E-mail: `jtl3@lehigh.edu`
Web: `http://www.lehigh.edu/~jtl3`

This course will introduce students to the principles involved in designing, analyzing, and implementing basic algorithms common in systems engineering applications.

Required Text

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*, Second Edition (McGraw-Hill, 2003).

There are a number of very good algorithms books out there. We will be using the Java language to implement the algorithms we will learn about, so books that also contain refreshers on the Java language may also be useful.

Recommended Texts

Donald Knuth. *The Art of Computer Programming. Volumes 1-3*, Third Edition (Addison-Wesley, 1997). *This is the* **bible** *of algorithm design and analysis. It was named one of the 12 best physical science monographs of the 20th century.*

Robert Sedgewick. *Algorithms in Java*, Third edition (Addison-Wesley, 2003).

Elliot B. Koffman and Paul A. T. Wolfgang . *Objects, Abstraction, Data Structures, and Design Using Java Version 5.0* (John Wiley and Sons, 2005).

David Flanagan. *Java in a Nutshell*, 5th Edition (O'Reilly, 1999). *This book is a good reference for all things Java*

Overview

This course will introduce students to the principles involved in designing, analyzing, and implementing basic algorithms common in systems engineering applications. The course will be divided into four units by topic area (see detailed syllabus below). Course meetings will consist of three 50 minute lectures and one three hour laboratory each week. The laboratory exercises will consist of case studies in which the students will have to apply the principles discussed during the week's lectures to solve a given systems engineering problem. This will be accomplished mainly through implementing various algorithms and data structures in Java. Weekly homework based on the laboratory case studies will further reinforce student learning.

Course Objectives

The goals of this course are for each student to:

1. Understand the basic principles of algorithm design, especially for applications in systems engineering;
2. Learn (discrete) mathematical fundamentals necessary for algorithmic analysis;
3. Understand basic techniques for analyzing the performance of algorithms;
4. Develop an appreciation for the importance of implementing algorithms efficiently and the skills necessary for doing so;
5. Develop an ability to solve systems engineering problems by designing and implementing an appropriate algorithm.
6. Sharpen and extend basic computing and programming skills
7. Hone techniques for debugging programs via hands-on experience
8. Develop deeper knowledge of advanced data structures and Java frameworks implementing these data structures
9. Learn how one might implement fundamental Collections types, such as List and HashSet
10. Learn various well-known searching and sorting algorithms
11. Discover methods for representing discrete structures such as graphs and basic operations on graphs
12. Learn and implement well-known algorithms for solving optimization problems on graphs
13. Appreciate the importance of numerical methods through implementing core techniques such as solving linear systems and solving least squares problems

Requirements and Grading

Prerequisites

Students are expected to have basic computing skills, including familiarity with the Java Programming language. They should also have completed ENG 1 and CSc 17.

Reading

Students will be expected to complete the readings associated with each lecture. Most readings will be from the course text, but students are encouraged to seek supplementary material. Links to supplementary reading material available over the Web will be available from the course page.

Lectures

Practical problem solving involves more than the ability to analyze and model real-world situations. Communication, collaboration, and team-work are key ingredients. As such, lectures will be interactive, and a portion the grade will depend on participation.

Labs and Problem Sets

Students will be required to attend laboratories each week and compose a laboratory report due one week after the laboratory occurs. Details on the structure of the laboratories will be given subsequently. Students will be required to complete weekly homework assignments related to the laboratories. The homework assignments will be submitted in conjunction with the laboratory reports each week.

Students are encouraged to discuss problems, but each student must write up his/her homework individually. In addition, students must cite with whom they have collaborated as well as external references used. I encourage collaboration and the use of reference materials. Indeed, these are key components of practical problem solving. However, you *must* site collaborators and references. Failure to cite references, and other types of cheating, will result in a serious grade penalty. Also, like many things, too much collaboration is not necessarily a good thing. If I think during the course of the semester that by collaborating, you are not learning the material, I will curtail the amount of collaboration that is allowed.

Exams

The emphasis of the course will be on the laboratories. However, to best gauge individual achievement and retention of the course material, a number of quizzes will be given. Depending on the progress of the course, the quizzes may be in-class 50 minute quizzes. Or 3 hour lab-based quizzes. There will be a final exam as well.

Academic Integrity

You are encouraged to share ideas with each other on class assignments **orally**. However, you must ultimately demonstrate your understanding of the material by writing up your own solutions without the help of other students or their written work, including source code. **AT NO TIME ARE YOU ALLOWED TO OBTAIN, LOOK AT, OR CUT AND PASTE FROM THE SOURCE CODE OR LABORATORY WRITE-UP/PROBLEM SET OF ANY OTHER INDIVIDUAL, WHETHER THEY ARE IN THE CLASS THIS SEMESTER**

OR NOT. THIS INCLUDES GETTING SOURCE CODE FROM THE INTERNET. YOU ARE EXPRESSLY FORBIDDEN FROM SHARING ELECTRONIC FILES OF ANY TYPE BY ANY MEANS WITH ANYONE.

Use of External Sources

Extensive materials related to the topics we will be covering in this class are available on the Internet. This may include, at times, source code for algorithms we will be studying. **WITH THE EXCEPTION OF SOURCE CODE**, you may use supplementary material to enhance your understanding of the course material. However, you should not look at other people's source code and you should not cut and paste from any source on the Internet. You will learn a great deal more if you complete the assignments on your own before consulting external references. If you don't understand how to solve the laboratory assignments, you will not do well on the exams.

If you use external references of any kind, even if they are not quoted verbatim, **YOU MUST CITE THEM**! This rule will be strictly enforced and violations will be dealt with harshly. Again, I encourage you to seek outside sources, but **you must acknowledge the source of any ideas that are not your own**.

You are all grown-ups. **Do not cheat**. If you have any question or concern about what constitutes cheating or improper collaboration, *please* contact me. An excellent web site with lots of useful information about the integrity policy and procedures at Lehigh is `http://www.lehigh.edu/~inprv/academicintegrity.html`. If I suspect that you are cheating, you will make me sad. Then you will make me mad. Do not do this. If you don't think that I will know if you are copying or inappropriately using other people's work to do your assignments, you are wrong.

Accommodations for Students with Disabilities

If you have a disability for which you are or may be requesting accommodations, please contact both your instructor and the Office of Academic Support Services, University Center 212 (610-758-4152) as early as possible in the semester. You must have documentation from the Academic Support Services office before accommodations can be granted.

Grading Scheme

- 10% Participation
- 30% Labs and Problem Sets
- 30% Quizzes
- 30% Final Exam

There is a penalty of 10% for each day that an assignment is late, *without exception.* Once an assignment has been graded and returned, homework will no longer be accepted.

(Tentative) Chronological Syllabus

There is some flexibility in the schedule, as I will speed up or slow down based on how well I feel we are retaining the course material.

Part I: **Introduction to Algorithm Analysis** *Jan. 15–Jan.26*

Basic introduction to the principles of algorithm design and analysis.

Readings: CLRS, Chapters 1–4, 10

Part II: **Sorting and Searching** *Jan. 29–Feb. 23*

Merge Sort, Insertion Sort. Heaps and Heapsort. Priority queues. Lists and Hashes. Dynamic Programming

Readings: CLRS Chapters 6, 11, 12, 15

Part III: **Graph Algorithms** *Feb. 26–Mar. 30*

Graphs and their representations. Basic graph algorithms. Spanning tree, shortest path, and max flow.

Readings: CLRS Chapters 22-26

Part IV: **Numerical Algorithms** *Apr.2–Apr. 27*

Matrix Properties. Sparse Matrices. Solving Linear Systems and Least Squares.

Readings: CLRS Chapter 28 and handouts