Data Structures

CMSC 420 - Fall 2007

COURSE DESCRIPTION

I. *Objectives.*

The main aim of the course is to familiarize the student with the fundamentals of data structures and with well-known techniques for manipulating data structures. The student will learn to apply these methods to applications in data base management systems, artificial intelligence, programming language implementation, operating systems, game programming, computer graphics, computational geometry, VLSI design, image processing, and computer vision.

II. *Topics to be Covered.*

- Basic Data Structures (2 review lectures)
  - Stacks, Queues, and Deques
  - Sequential Allocation
  - Linked Allocation and Topological Sorting
  - Circular Lists
  - Doubly-Linked Lists
  - Arrays
- Trees (2 review lectures)
  - Terminology
  - Traversals
  - Binary Tree Representation of Trees
  - Other Representations of Trees
  - Processing Equivalences with UNION-FIND
- Graphs (1 lecture)
- Winged Edge Data Structure (1 lecture)
- Sorting (2 lectures)
  - Review of methods
  - Heapsort
  - Quicksort
  - External Sorting
- Searching (1 lecture)

  - Sequential Searching
  - Binary Searching
  - Digital Searching
- Balanced-Tree Searching (1 lecture)
  - AVL Trees
  - Splay Trees
  - Skip Lists
- B-trees and Red-Black Trees (1 lecture)
- Introduction to LISP (6 lectures)
- Lists and Garbage Collection (1 lecture)
- Dynamic Storage Allocation (1 lecture)
- Point Methods (2 lectures)
  - Quadtrees
  - K-d Trees
  - Grid File
  - EXCELL
- Hashing (2 lectures)
  - Hashing Functions
  - Chaining
  - Open Addressing
  - Brent's Method
  - Comparison
- Alternative Rectangle Representations (1 lecture)
  - Quadtree Approaches
  - R-trees
- Priority Search Trees and Range Trees (1 lecture)
- Representations of Line Segments (1 lecture)

Some changes to this list of topics, and the depth and order of their coverage may take place as the course proceeds and due to the difference between the length of the Fall and Spring semesters as well as the interests and backgrounds of the students.

III. *Homework.*

- 4-8 homework assignments consisting of problems.
- 2 short LISP written assignments

The due date for each assignment will be specified. Late homework will NOT be accepted.

IV. *Projects.*

- A major programming project (Project 1) to be written in C, C++, or PASCAL. It will be submitted incrementally as specified by the instructor. The project will be worth 80 points in total. For this project you will need to read a particular set of pages in [1].
- A LISP warm up assignment (Project 2) to familiarize yourself with the LISP system. This project will be worth 15 points and counts like a homework assignment.
- 3 LISP programming assignments that should not require more than 80 lines of code apiece. Project 3 will be worth 30 points. Project 4 will be worth 20 points. Project 5 will be worth 30 points.

The explanations will be in terms of a simple variant of LISP that does not use parentheses. The due date for each project will be specified. Late projects will generally NOT be accepted. If lateness is permitted, then notice will be given as to the penalty which will be a deduction of a given number of points per day the project is late.

V. *Exams.*

Two examinations will be held. The midterm will be either on Tuesday, October 23, or Thursday, October 18, with the exact date being announced in class. The second exam will be on Monday, December 17 at 10:30AM-12:30PM. The second exam is not cumulative. It covers material since the midterm.

All exams are closed book and closed notes. If there are any changes to the dates of the exams, they will be announced in class and posted on the web.

VI. *Grading.*

Grades will be calculated in terms of the student's performance with the following approximate weightings:

1. Mid-Term I $(20 - 30\%)$
2. Final Exam $(20 - 30\%)$
3. Projects $(30 - 40\%)$
4. Homework $(10 - 20\%)$

If you are a graduate student in Computer Science and you wish for this course to count as one of the 7 courses for the new qualifying exam, then you need to speak to Professor Samet and do the optional parts of the projects as well as possibly some additional homework problems and operations on the projects.

VII. *Texts.*

The required texts are:

1. H. Samet. "Foundations of Multidimensional and Metric Data Structures", Morgan-Kaufmann publishers, San Francisco, CA, Aug 2006.
2. H. Samet. "Notes on Data Structures", University of Maryland, College Park, MD, 2003 (available in lecture note form for purchase at the University Book Center).

The recommended texts are:

1. D. E. Knuth. "The Art of Computer Programming", vol. 1, "Fundamental Algorithms", Third Edition, Addison-Wesley, Reading, MA, 1997.
2. D.E. Knuth. "The Art of Computer Programming", vol. 3, "Sorting and Searching", Second Edition, Addison-Wesley, Reading, MA, 1998.
3. D. P. Friedman and M. Felleisen. "The Little LISPer", Third Edition, Macmillan, New York, 1989.

Additional references are:

1. P. H. Winston and B. K. P. Horn. "LISP", Third Edition, Addison-Wesley, Reading, MA, 1989.
2. H. Samet. "Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS", Addison Wesley, Reading, MA, 1990.

VIII. *Instructor.*

Prof. H. Samet. Office Hours: Tuesday and Thursday in 4425 A.V. Williams Building, 1:00-2:00PM. Telephone: (301) 405-1755 Questions pertaining to the projects and homework assignments should be directed to the Teaching Assistant.

IX. *Teaching Assistant.*

Mike Lieberman Office Hours: TBA, 4431 A.V. Williams Building

X. *Miscellaneous.*

1. All graded materials (examinations and programming assignments and homework) must be strictly individual efforts. Cooperation on homework programming assignments is limited to general discussion of the problem (not its solution), and assistance with errors. Additional cooperation is considered academic dishonesty. Transmitting a copy of a solution (in either hardcopy or electronic form), falsely representing the correctness of a program or homework, or delaying other members of the class from completing a programming assignment are considered forms of academic dishonesty.

2. Students claiming an excused absence must apply in writing and furnish documentary support (such as from a health care professional who treated the student) for any assertion that the absence qualifies as an excused absence. The support should explicitly indicate the dates or times the student was incapacitated due to illness. Self-documentation of illness is not itself sufficient support to excuse the absence. An instructor is not under obligation to offer a substitute assignment or to give a student a make-up assessment unless the failure to perform was due to an excused absence. An excused absence for an individual typically does not translate into an extension for team deliverables on a project.

3. Any student eligible for and requesting reasonable academic accommodations due to a disability is requested to provide, to the instructor in office hours, a letter of accommodation from the Office of Disability Support Services (DSS) within the first two weeks of the semester.