Data Structures
CMSC 420 - Fall 2003

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, 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 (Sections 1.1 and 1.2 in [2] or [REC_1] 234-238)
  - Sequential Allocation (Section 1.2 in [2] or [REC_1] 240-248)
  - Linked Allocation and Topological Sorting (Section 1.3 in [2] or [REC_1] 251-263)
  - Circular Lists (Section 1.2 in [2] or [REC_1] 270-272)
  - Doubly-Linked Lists (Section 1.2 in [2] or [REC_1] 278-279)
  - Arrays (Section 1.4 in [2] or [REC_1] 295-304)

- Trees (2 review lectures)
  - Terminology (Section 1.5 in [2] or [REC_1] 305-315)
  - Traversals (Section 1.5 in [2] or [REC_1] 315-332)
  - Binary Tree Representation of Trees (Section 1.5 in [2] or [REC_1] 332-336)
  - Other Representations of Trees (Section 1.5 in [2] or [REC_1] 347-352)
  - Processing Equivalences with UNION-FIND (Section 1.5 in [2] or [REC_1] 353-355)

- Graphs (1 lecture) (Section 1.6 in [2] or [REC_1] 362-364, 371-373, 399-405)

- Winged Edge Data Structure (1 lecture) (Section 1.7 in [1] 331-335)

- Sorting (2 lectures)
  - Review of methods (Section 4.1 in [2] or [REC_2] 76-84, 106-107)
  - Heapsort (Section 4.2 in [2] or [REC_2] 142-147, 150-153)
  - Quicksort (Section 4.3 in [2] or [REC_2] 114-125)
  - External Sorting (Sections 4.3 and 4.4 in [2] or [REC_2] 247-258, 266-269)

- Searching (1 lecture)
- Sequential Searching (Section 5.1 in [2] or [REC,2] 393-401)
- Binary Searching (Section 5.1 in [2] or [REC,2] 406-407)
- Digital Searching (Section 5.3 in [2] or [REC,2] 481-493)

- Balanced-Tree Searching (1 lecture)
  - AVL Trees (Section 5.1 in [2])
  - Splay Trees (Section 5.1 in [2])
  - Skip Lists (Section 5.1 in [2])

- B-trees and Red-Black Trees (1 lecture) (Section 5.2 in [2] or [REC,2] 471-480)

- Introduction to LISP (6 lectures) (Chapter 2 in [2] or [REC,3,REC,4])

- Lists and Garbage Collection (1 lecture) (Section 3.2 in [2] or [REC,1] 406-422)

- Dynamic Storage Allocation (1 lecture) (Section 3.1 and 3.3 in [2] or [REC,1] 435-455)

- Point Methods (2 lectures)
  - Quadtree ([1] 1-10, 43-65, 85-105)
  - K-d Trees ([1] 66-80)
  - Grid File ([1] 135-141)
  - EXCELL ([1] 141-151)

- Hashing (2 lectures)
  - Hashing Functions (Section 6.1 in [2] or [REC,2] 506-509)
  - Chaining (Section 6.1 in [2] or [REC,2] 513-515)
  - Open Addressing (Section 6.1 in [2] or [REC,2] 518-522)
  - Brent’s Method (Section 6.1 in [2] or [REC,2] 525-527)
  - Comparison (Section 6.4 in [2] or [REC,2] 538-542)

- Alternative Rectangle Representations (1 lecture)
  - Quadtree Approaches ([1] 200-219)
  - R-trees ([1] 219-225)


- Representations of Line Segments (1 lecture) ([1] 227-264)

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 warmup 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 21, or Thursday, October 23, with the exact date being announced in class. The second exam will be on Thursday, December 18 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:


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:


Additional references are:


VIII. **Instructor.**

Prof. H. Samet. Office Hours: Tuesday 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.**

Konstantinos Bitsakos Office Hours: Monday and Wednesday in 1151 A.V. Williams Building, 2PM.
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. If you have a documented disability and wish to discuss academic accommodations with me, please contact me as soon as possible.