Data Structures

CMSC 420 - Spring 1997

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 lectures)
  - Stacks, Queues, and Deques (Sections 1.1 and 1.2 in [1] or [REC_1] 234-238)
  - Sequential Allocation (Section 1.2 in [1] or [REC_1] 240-248)
  - Circular Lists (Section 1.2 in [1] or [REC_1] 270-272)
  - Doubly-Linked Lists (Section 1.2 in [1] or [REC_1] 278-279)
  - Linked Allocation and Topological Sorting (Section 1.3 in [1] or [REC_1]251-263)
  - Arrays (Section 1.4 in [1] or [REC_1] 295-304)
- Trees (2 lectures)
  - Terminology (Section 1.5 in [1] or [REC_1] 305-315)
  - Traversals (Section 1.5 in [1] or [REC_1] 315-332)
  - Binary Tree Representation of Trees (Section 1.5 in [1] or [REC_1] 332-336)
  - Other Representations of Trees (Section 1.5 in [1] or [REC_1] 347-352)
  - Processing Equivalences with UNION-FIND (Section 1.5 in [1] or [REC_1] 353-355)
- Graphs (1 lecture) (Section 1.6 in [1] or [REC_1] 362-364, 371-373, 399-405)
- LISP Fundamentals and Recursive Algorithms (5 lectures) (Chapter 2 in [1] or [REC_3,REC_4])
- Lists and Garbage Collection (1 lecture) (Section 3.2 in [1] or [REC_1] 406-422)
- Dynamic Storage Allocation (1 lecture) (Section 3.1 and 3.3 in [1] or [REC_1] 435-455)
- Sorting (2 lectures)
  - Review of methods (Section 4.1 in [1] or [REC_2] 76-84, 106-107)
  - Heapsort (Section 4.2 in [1] or [REC_2] 142-147, 150-153)
  - Quicksort (Section 4.3 in [1] or [REC_2] 114-125)
  - External Sorting (Sections 4.3 and 4.4 in [1] or [REC_2] 247-258, 266-269)
- Searching (1 lecture)
  - Sequential Searching (Section 5.1 in [1] or [REC_2] 393-401)
  - Binary Searching (Section 5.1 in [1] or [REC_2] 406-407)
  - Digital Searching (Section 5.3 in [1] or [REC_2] 481-493)

- Balanced-Tree Searching (1 lecture)
  - AVL Trees (Section 5.1 in [1])
  - Splay Trees (Section 5.1 in [1])
  - Skip Lists (Section 5.1 in [1])
- B-trees and Red-Black Trees (1 lecture) (Section 5.2 in [1] or [REC_2] 471-480)
- Hashing (2 lectures)
  - Hashing Functions (Section 6.1 in [1] or [REC_2] 506-509)
  - Chaining (Section 6.1 in [1] or [REC_2] 513-515)
  - Open Addressing (Section 6.1 in [1] or [REC_2] 518-522)
  - Brent's Method (Section 6.1 in [1] or [REC_2] 525-527)
  - Linear Hashing ([2] 116-124 or Section 6.2 in [1])
  - Spiral Hashing ([2] 125-135 or Section 6.3 in [1])
  - Comparison (Section 6.4 in [1] or [REC_2] 538-542)
- Point Methods (2 lectures)
  - Quadtrees ([2] 1-10, 43-65, 85-105)
  - K-d Trees ([2] 66-80)
  - Grid File ([2] 135-141)
  - EXCELL ([2] 141-151)
- Rectangle Representations (2 lectures)
  - Plane-Sweep, Segment Trees, and Interval Trees ([2] 153-174)
  - Quadtree Approaches ([2] 200-219)
  - R-trees ([2] 219-225)
  - Priority Search Trees and Range Trees ([2] 80-85, 171-174)
- Other Spatial Structures (1 lecture)
  - Winged Edge Data Structure ([1] Section 1.7, [2] 331-335)
  - Representations of Line Segments ([2] 227-264)

Some changes to this list may take place as the course proceeds.

III. *Assignments.*

- 6-10 homework assignments consisting of written problems.
- A major programming project to be written in PASCAL, C, or C++. 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 [2].
- 3-5 other small programming problems, including some in lisp.
- The due date for each assignment will be specified in class and on the web page.

IV. *Exams.*

Two examinations will be held. The midterm is tentatively scheduled for April 1st. If there is a change, it will be announced in class. The final exam will be on Thursday, May 15th. The final is not cumulative. It covers material since the midterm.

V. *Grading.*

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

1. Mid-Term Exam (20%)
2. Final Exam (30%)
3. Programming Projects (30%)
4. Written Assignments (20%)

VI. *Texts.*

The required texts are:

1. H. Samet. "Notes on Data Structures", University of Maryland, College Park, MD, 1997 (will be available in the University Copy Center for purchase at the cost of copying to each student in the class).
2. H. Samet. "The Design and Analysis of Spatial Data Structures", Addison Wesley, Reading, MA, 1990.

The recommended texts are:

1. D. E. Knuth. "The Art of Computer Programming", vol. 1, "Fundamental Algorithms", Second Edition, Addison-Wesley, Reading, MA, 1973.
2. D.E. Knuth. "The Art of Computer Programming", vol. 3, "Sorting and Searching", Addison-Wesley, Reading, MA, 1973.
3. D. P. Friedman and M. Felleisen. "The Little LISPer", Third Edition, Macmillan, New York, 1989.
4. C. J. Rieger III. "Everyman's LISP", University of Maryland, College Park, MD, 1974.

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.

VII. *Other Materials.*

- Copies of the overhead projections used in the lectures will be made available the week before class, on the webpage and in the main course account (see below). You are encouraged to print these out before class and use them to help take notes.
- In order to complete and hand in the programming assignments, computer accounts will be provided on aITS cluster machines (e.g., marlowe.umd.edu, marple.umd.edu, tracy.umd.edu). Please see the TA for further details. Also make sure to obey the acceptable use guidelines for these accounts. Course information (including this syllabus, and assignments and lecture notes) will be made available in the directory of account dt01 (from your home directory: `cd ../dt01`). Please do not make your own directories publicly readable.
- A Webpage for this course has been set up at `http://www.cs.umd.edu/ traum/CMSC420/` this will contain pointers to the course materials, as well as updated schedules.

VIII. *Staff*

	Instructor	**TA**
	Dr. David R. Traum	Kyongil Yoon
Office	3161 A.V. Williams Building	Room 1109 AVW
Office Hours	Thursday 1:00-3:00pm	Tuesday 3:00-5:00pm
	or by appointment.	Friday 9-11am
Telephone	(301) 405-1139	(301) 405-2715
email	traum@cs.umd.edu	kiyoon@cs.umd.edu

Questions pertaining to the projects and homework assignments should be directed to the Teaching Assistant.

IX. *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. See the Undergraduate Catalog for definitions and sanctions. Academic dishonesty is a serious offense which may result in suspension or expulsion from the University. 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, please contact the instructor as soon as possible.