Data Structures

CMSC 420 - Fall 2020

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, spatial databases, 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 not necessarily in this order*

- 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)
- Lists and Garbage Collection (1 lecture)
- Dynamic Storage Allocation (1 lecture)
- Point Methods (2 lectures)
  - Quadtrees
  - K-d Trees
  - Grid File
  - EXCELL
- Hashing (4 lectures)
  - Hashing Functions
  - Chaining
  - Open Addressing
  - Brent's Method
  - Comparison
  - Linear Hashing ([1] 116-124 or Section 6.2 in [2])
  - Spiral Hashing ([1] 125-135 or Section 6.3 in [2])
- Alternative Rectangle Representations (2 lecture)
  - Plane-Sweep, Segment Trees, and Interval Trees ([1] 153-174) (1 lecture)
  - Quadtree Approaches
  - R-trees
- Representations of Line Segments (1 lecture)
- Incremental nearest neighbor and approximate (1 lecture)
- Priority Search Trees and Range Trees (1 lecture)
- Introduction to LISP data structures (1 lecture)
- LISP data structure exercise (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, backgrounds, and skill levels 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. It will be submitted incrementally as specified by the instructor. The project will be worth approximately 100 points in total. For this project you will need to read a particular set of pages in [1]. The project will be distributed (posted) before the second class meeting with the first two parts being due by the end of the second week of the semester. The amount of work involved in it varies depending on the student's comfort level with the use of pointers. As this may be substantial for some students, the project is to be completed in parts by the last week of the semester

The due date for each part of the 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 class meeting day that the project is late.

V. *Exams.*

One exam will be held most likely just before Thanksgiving. The exact date will be posted once we know for sure of the plans of the campus.

VI. *Grading.*

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

1. Exam (20 – 30%)
2. Projects (50 – 60%)
3. Homework (20 – 30%)

A curve will be used which is based in part on prior instances of the class so that we have historically consistent grading. Also if a student does really well on the final and projects are completed well, then a very poor exam score may be disregarded.

If you are a graduate student in Computer Science and you wish for this course to count as one of the 6 courses for the Ph.D. qualifying exam, then you need to speak to Professor Samet and do the optional parts of the project (if any) as well as possibly some additional homework problems and operations on the projects.

Good Faith Attempt: You must satisfy a minimum set of requirements for each project (Good Faith Attempt) otherwise you will not pass the course (automatic grade of F). Each part of the project defines its own good faith attempt criteria and a deadline to provide an implementation that satisfies it. If you start the parts of the project on time, and look for assistance (if required), then you should have no problems satisfying the Good Faith Attempt. The Good Faith Attempt requirement guarantees to prospective employers that you have the skills implied by their inclusion in your transcript as upper-level courses in the UMD degree program. Notice that you will not receive extra points for completing the good faith attempt. The grade you obtain for a project will be based on your ontime/late submission.

VII. *Texts.*

The required texts are:

1. H. Samet. "Foundations of Multidimensional and Metric Data Structures", Morgan-Kaufmann publishers, San Francisco, CA, Aug 2006. You can purchase it at the University Book Center, at Amazon.com currently at $58.76 at the Elsevier web site if you enter the promotion code COMP320 which reflects a 30% discount. There was also a code ATRSCHOOL which yielded a 40% discount if you acted quickly. In any case, when buying at the Elsevier web site, the shipping is free, while Amazon may charge for shipping if you do not have Amazon Prime.

2. H. Samet. "Notes on Data Structures", University of Maryland, College Park, MD, 2016. It will be posted for free on the class web pages. They are also available in lecture note form for purchase for $25 at the Engineering Copy Center which is in the Engineering Building Room 1123 near the Food Court. The lectures follow the notes very closely and thus you may not need to take notes. You may want to call ahead (301) 405-3875 to make sure that a copy is available for you when you go there. http://www.eng.umd.edu/copycenter/index.html

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.

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: Tuesdays 11AM to 12 noon in IRB 4252 or online. Telephone: (301) 405-1755 Questions pertaining to the projects and homework assignments should be directed to the Teaching Assistant.

IX. *Teaching Assistants.*

Yunheng Han e-mail yhhan@terpmail.umd.edu Office Hours: Monday and Wednesday from 10AM to 12 noon online.

Mackenzie Kong-Sivert e-mail: mkongsiv@umd.edu

Office Hours: Monday from 9AM to 11AM and Tuesday from 3:30PM to 4:30PM online.

X. *Miscellaneous.*

1. All graded materials (examinations and programming assignments and homework) must be strictly individual efforts. Cooperation on homework programming assignments (unless explicitly indicated otherwise) is limited to general discussion of the problem (not its solution), and assistance with errors. Additional cooperation is considered academic dishonesty and is a violation of the Code of Academic Integrity. Remember that the purpose of projects is for you to learn the class material and do well in projects. 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. Similarly, posting project solutions in a public online location is also considered a form of academic dishonesty.

2. See http://www.ugst.umd.edu/courserelatedpolicies.html for extensive university course policy information.

3. Any student who needs to be excused for an absence from a single lecture, recitation, or lab due to a medically necessitated absence shall: a) Make a reasonable attempt to inform the instructor of his/her illness prior to the class. b) Upon returning to the class, present their instructor with a self-signed note attesting to the date of their illness. Each note must contain an acknowledgment by the student that the information provided is true and correct. Providing false information to University officials is prohibited under Part 9(i) of the Code of Student Conduct (V-1.00(B) University of Maryland Code of Student Conduct) and may result in disciplinary action.

   The self-documentation may not be used for the Major Scheduled Grading Events (midterm and final exams and programming project due dates) and it may only be used for only 1 class meeting during the semester. Any student who needs to be excused for a prolonged absence (2 or more consecutive class meetings), or for a Major Scheduled Grading Event, must provide written documentation of the illness from the Health Center or from an outside health care provider. This documentation must verify dates of treatment and indicate the timeframe that the student was unable to meet academic responsibilities. In addition, it must contain the name and phone number of the medical service provider to be used if verification is needed. No diagnostic information will ever be requested.

4. The instructor is not under obligation to offer a substitute assignment or to give a student a make-up assessment for missing a Major Scheduled Grading Events unless the failure to perform was due to an excused absence. A valid excused absence will need to be granted through the student presenting documentation from the Health Center or from an outside health care professional. This documentation must be submitted within one week of returning to classes and must include dates of incapacitation as well as the name and phone number of the health care provider. No diagnostic information shall be given. Once the period of incapacitation is over, the student must meet the missed academic responsibilities at according to the requirements and specifications set forth by the instructor. The Major Scheduled Grading Events for this course include:

  - Exam: Most likely on Thursday during the week before the Thanksgiving week which depends on the campus closing policy. The date will be posted once more information becomes available.
  - Programming Project Due Dates throughout the semester.

5. 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.

6. Requests for retroactive accommodation are not considered reasonable. In the same vein we do not consider it reasonable to ask an instructor to create an alternate assignment of substance. The spirit of our accommodation should be to help DSS-advised students find creative ways to meet the high standards we set for all our students.