CMSC 420
Larry Davis
X56718
Class - TT 12:30-1:45
Office TT 2:00-3:00 and by appt.

Secretary: Ms. Becky Flick
301-405-6722

email: lsd@umiacs.umd.edu
http://www.umiacs.umd.edu/users/lsd/420.html

Texts and grading

Texts:NONE

Recommended reading from:

1) Data structures and algorithms in Java: Goodrich and Tammasia

2) Notes on Data Structures: Samet

These will be put on reserve in the Engineering Library

Grading:

1) 1 exam in class + final - 70%

2) 2 projects - 30%

Late projects subject to penalties (to be announced)

The syllabus

- Check the web page for an up-to-date syllabus
  - Notes are available for downloading - check they are revised for Fall 1998
- Check the web page for project descriptions
  - dates, data, penalties, FAQS
- Check the web page for exam information
  - old exams (sometimes with answers)
  - dates
  - questions you should prepare ahead of time!
- Check the web page for pointers to other sites that may help you understand the course material

Schedule

- No classes on the following dates:
  - November 3,5
  - November 24 (Hooray!)
- First (midterm) exam scheduled for October 8, to cover material up to October 1.
- Final exam determined by University scheduling (think it is December 19)

What you should already know

- **Basic data structures**
  - **Stacks** - push, pop, sequential allocation, linked allocation, role in evaluation of arithmetic expressions
  - **Queues, Dequeues** - insertion and deletion, doubly linked implementations
  - **Trees** - binary trees, tree traversals - preorder, inorder and postorder - recursive and nonrecursive algorithms for tree traversals, storing and searching dictionaries in binary trees
- **Basic algorithms**
  - **Search** - binary search, binary tree search, sequential search
  - **Sorting** - quicksort, treesort, elementary quadratic sorts
- Basic O() characterization of algorithm complexity
- C/C++ for projects (almost definite)

What is this course about?

- Fundamental data structures and operations that arise in core areas of computer science
  - compilers
  - operating systems
  - database systems
    - » relational
    - » spatial
- For all data structures we study
  - insertion
  - deletion
  - lookup
- Other typical operations
  - merging
  - traversals

Applications of data structures

- Compilers
  - How are symbol tables organized for fast access?
    - » every time a variable is encountered during compilation it must be checked for type, whether it has been assigned a value
    - » every array reference must be resolved to a physical address, and bounds must be checked
  - How are sequential and linked data structures represented by compilers for fast run-time execution?

Applications of data structures

- Operating systems
  - How are system resources, like memory, represented so that they can be efficiently shared amongst processes and users?
  - What data structures are needed to support sophisticated GUI's?
- Geographic information systems
  - How are maps represented in a computer?
  - How can maps be combined and displayed?

Applications of data structures

- Computer vision and image processing
  - How are images represented in a computer?
  - How can we recognize the contents of an image from its representation?
- Virtual reality
  - How can a complete 3-D world be represented in a computer?
  - How can we generate real-time image sequences as a user navigates through such a world?

Pedagogic approach

- Define a concrete problem
  - represent a highway map for route planning
- Develop an “abstraction” of the problem
  - design a data structure for representing collections of points and lines in the Cartesian plane
- Propose alternative data structures for solving the problem
  - binary arrays, graph structures, quadtrees
- Investigate the properties of these data structures
  - time/space tradeoffs
- Do a couple of projects!
- This is NOT a theory course, but we still need to analyze algorithms