COMPUTER SCIENCE 361

ALGORITHMS & DATA STRUCTURES

I. Introduction

A. Catalog Description

This is a course in advanced data structures, the algorithms needed to manipulate these data structures, proofs that the algorithms are correct, and a runtime analysis of the algorithms. Students study advanced data structures such as Red-Black Trees, 2-3 Trees, Heaps and Graphs. Students also study algorithm design techniques including Greedy Algorithms, Divide and Conquer, Dynamic Programming, and Backtracking. They also learn about NP-Complete problems. Satisfies a writing requirement in major contracts. *Prerequisites: CSCI 261, CSCI 281 (may be taken concurrently), and either MATH 210 or MATH 290 (MATH 290 may be taken concurrently).* Offered Spring term only.

B. Learning Objectives

Students will learn about advanced data structures, the algorithms needed to manipulate those data structures, and the tools for the analysis of those algorithms. CSCI361 extends the knowledge the student gained in CSCI 261 (Computer Science II) to a more mathematical description of data structures and algorithms.

C. Prerequisites

CSCI 261, CSCI 281 (may be taken concurrently) and Math 210 or Math 290 (Math 290 may be taken concurrently). A grade of C- or better is required in prerequisite courses.

II. Required Topics

A. Algorithm Design Techniques

1. Divide and conquer
2. Dynamic programming
3. Greedy algorithms
4. Backtracking

B. Searching and sorting

1. At least one of red-black trees or 2-3 trees
2. Binary, Leftist and Skew Heaps
3. Hash tables
4. Selection
5. Quicksort, Heapsort, and Shellsort
6. Linear time sorting (radix and counting/bucket sort)

II. Required Topics (cont.)

C. Graphs and Graph Algorithm

1. Depth-first and Breadth-first Search
2. Prim and Kruskal's algorithm for minimal spanning trees
3. Warshall and Dijkstra's shortest-path algorithms

D. Algorithm Analysis

1. Asymptotic bounds (Big-O, Big-Theta, Big-Omega, little-o, and little-omega)
2. Adversary arguments
3. Amortized analysis

E. NP-Complete Problems

1. Definition of P and NP
2. Polynomial reduction
3. Approximation algorithms

III. Optional Topics

A. String matching
B. String Matching and Text Similarity
C. Network Flow
D. Cryptography
E. Computational Geometry
F. External Memory Algorithms
G. Parallel Algorithms
H. Evaluating polynomials
I. Fast Fourier Transform
J. Splay Trees and Skip Lists

IV. Bibliography

Aho, Hopcroft, & Ullman, Data Structures and Algorithms
Basse and Gelder, Computer Algorithms
Cormen, et. al. Introduction to Algorithms
David Gries, Science of Programming
Donald Knuth, The Art of Computer Programming, Vol. I, II, III
Goodrich & Tamassia Algorithm Design: Foundations, Analysis, and Internet
Horowitz, Sahni, & Mehta, Fundamentals of Data Structures in C++

IV. Bibliography (cont.)

Kleinberg & Tardos	Algorithm Design
Thomas Standish,	Data Structure Techniques
Mark Allen Weiss,	Data Structures & Algorithm Analysis in Java
Nicklaus Wirth,	Algorithms + Data Structures = Programs
CD ROM	Dr. Dobb's Essential Books on Algorithms and Data Structures