Algorithms and Data | Spring, 2009
CS U690 | Professor G. Cooperman

Course Description:

Niklaus Wirth famously wrote "programs = algorithms + data structures". This course is about developing algorithms and data structures. It is distinguished from programming in that we will work primarily with pseudo-code.

Note that progress in computer science would be greatly accelerated if we were able to completely eliminate the implementation stage. This would allow us to spend more time on improving the algorithms, while eliminating all the drudgery of the implementation stage: no debugging, no object-oriented programming, no software engineering. This course explores the benefits of such a utopia.

The course presents a tour of interesting algorithms and data structures that either continue to have a large impact or are likely to have a future large impact on computer science. Examples include: fast multiplication of polynomials of degree n (in time $O(n \log n)$ instead of in time $O(n^2)$); search (breadth-first and depth-first search); data compression (Huffman encoding, which is the second stage of the Deflate algorithm used by zip and gzip); and automation of logic (formal verification). Note that all other algorithms in computer science can be expressed as problems in logic. Its application in formal verification has been broadly successful in silicon circuit verification, and is now increasingly used for proving software correctness.

Faculty Information:

Professor G. Cooperman
Office: 336 West Village Hall
Phone: 617 373-8686
e-mail: gene@ccs.neu.edu
Office Hours: Mondays, 10:20 - 11:30; Wednesdays, 12:00 - 1:00; and by appointment.

Textbook:

The textbook for the course is:
Algorithms, by Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani; McGraw-Hill, 2006.

Exams and Grades:

The course grade will be based on homeworks, an in-class mid-term, and a take-home final exam. The midterm exam and final exam will contributing 30% and 40%, respectively, to your course grade. The remainder of the grade will be based on written homework assignments (30%). There will be a homework assignment during most weeks when there is no exam.

Syllabus:

Week	*Topics*	*Reading*
Jan. 5	Mathematical Preliminaries; Hashing, etc.	Prologue; Chap. 1
Jan. 12	Divide-And-Conquer	Chapter 2
Jan. 19	Divide-And-Conquer (fast poly. mult.)	Chapter 2
Jan. 26	Graph (BFS, DFS)	Chapters 3 – 4
Feb. 2	Graph (Dijkstra, priority queue)	Chapter 4
Feb. 9	Greedy Algorithms: Huffman enc., Horn formulas	Chapter 5
Feb. 16	Greedy Algorithms: Huffman enc., Horn formulas	Chapter 5
Feb. 23	Midterm; Greedy Algo. (cont.)	Chapter 6
Mar. 2	Dynamic Programming	Chapter 6
Mar. 9	Dynamic Programming	Chapter 6
Mar. 16	NP-Completeness	Chapters 8 – 9
Mar. 23	Non-polynomial time algos.	Chapters 8 – 9
Mar. 30	Non-poly. (formal verification)	Handouts
Apr. 6	More formal verification	Handouts
Apr. 13	Last Day of Classes	
Apr. 16 - 24	Final Exam	