CMSC 451: Design and Analysis of Algorithms
Fall 2000

Instructor: Clyde Kruskal. Office: AVW 3215. Office phone: 301-405–2683.
E-mail: `kruskal@cs.umd.edu`.
Office hours: Tuesday and Thursday: 1:30pm–3:00pm
Also by appointment.

Teaching Assistant: Chunqing Lin
Office hours: AVW 1151 (TA room), Wednesday 12:00pm-3:00pm
Office phone: 301-405-2775
E-mail (best way to contact): `cqlin@cs.umd.edu`
Also by appointment.

Course Overview: This course presents fundamental techniques for designing efficient computer algorithms, proving their correctness, and analyzing their complexity. General topics include graph algorithms, sorting, searching, and basic algorithm design paradigms (such as divide-and-conquer, dynamic programming, and greedy algorithms), lower bounds, and NP-completeness.

Text: Thomas Cormen, Charles Leiserson, Ron Rivest, *Introduction to Algorithms*, McGraw Hill and MIT Press, 1990.

Prerequisites: CMSC 214 and CMSC 251. Each student is expected to know the basic concepts of programming (e.g. loops, pointers, recursion), discrete mathematics (proof by induction, sets), simple data structures (lists, stacks, queues, trees), and calculus (logarithms, differentiation, integration).

Course Work: Course work will consist of written homework assignments and two exams (one midterm and a comprehensive final). Homework problems will be mathematically oriented. Homeworks are to be turned in at the start of class on the due date. All homeworks are to be written up independently. They should be clear and NEAT.

It is your responsibility to make sure that you pick up all homeworks and handouts.

Grading: Final grades will be based on homework assignments, the midterm exam, and the comprehensive final exam. The relative weights of these will be approximately 10% for the homework total, 40% for of the midterm, and 50% for the final exam.

Syllabus: The following is a tentative list of topics and readings in very approximate order. Topics in parentheses are likely to be omitted.

1. Sorting (Ch. 9)
2. Elementary graph algorithms (Ch. 23)
3. Dynamic programming (Ch. 16)
4. Greedy algorithms (Ch. 17)
5. Shortest paths (Ch.'s 25,26),
6. Minimum spanning trees (Ch. 24)
7. NP-completeness (Ch. 36)
8. Approximation algorithms (Ch. 37).