COMP 122 – Algorithms and Analysis

Spring 1999

Meeting Place: SN014
Meeting Time: Mon/Wed 12:30pm - 1:45pm

Instructor: Prof. Ming C. Lin
Telephone: 962-1974
Office: SN223
E-mail: lin@cs.unc.edu
Office Hours: Mon/Wed 11:00am - 12:00pm and by appointments

TA: Kok-Lim Low
Telephone: 962-1918
Office: SN 041
E-mail: lowk@cs.unc.edu
Office Hours: Wed 2:00-3:00pm and Thur 3:00pm-5:00pm

Text: *Introduction to Algorithms*, Cormen, Leiserson and Rivest, McGraw Hill, 1990.

Prerequisites: MATH 81 and COMP 121.

Course Description: This is an upper-division undergraduate course on the design and analysis of computer algorithms. This is a course on problem solving, not a programming course. While we will discuss implementation issues occasionally, there will be no programming associated with this course. We will examine several interesting problems, devise algorithms for solving them, prove their correctness, and characterize their performance. We will also study many useful data structures and core algorithms, different methods for designing algorithms to solve problems, various techniques for analyzing these algorithms, and applications of the algorithmics on practical problems in the real-world.
This course focuses primarily on developing thinking abilities on both formal thinking (proof techniques and algorithm analysis) and problem solving skills (algorithm design and application). Upon successful completion of the course, you should:

1. Be thoroughly familiar with a collection of *core algorithms*.
2. Be fluent in the following *algorithm design paradigms*: divide and conquer, greedy algorithms, randomization, dynamic programming and approximation methods .
3. Be able to analyze the correctness and runtime performance of a given algorithm using the following techniques: asymptotic notion, recurrences, proof by induction, and proof by contradiction.

4. Be familiar with the *inherent complexity* (lower bounds and intractability) of certain problems.
5. Be intimately familiar with the following data structures: lists, trees, graphs, heaps, balanced trees, hash tables, and disjoint sets.
6. Be able to *apply* these techniques in real-world problems.

Grading:		
	Quizzes	15%
	Midterm Exam	20%
	Final Exam	35%
	Homework	30%

Letter grades will not be assigned on the curve, but on absolute standards. Your final grade in this course will be determined solely by how much you learn. There is no predetermined grading scale. Numerical grades will be posted periodically throughout the semester. It is assumed to be the student's responsibility to make sure that the grades are accurately recorded. Please feel free to discuss your progress and your standing in this class with the instructor.

Examinations: The mid-semester exam will be given on Wednesday, March 17 during class. The comprehensive final exam will be held at 12:00 noon on the 7th of May 1999 in SN014. There will be a few quizzes covering either reading assignments or in-class materials, some announced in advance and some given without advanced notices. These in-class quizzes are designed to help reinforce your understanding of the course materials. All exams and quizzes will be closed book and closed notes.

Homework: Homework assignments are due *at the beginning of each class* on the due date given. *No late homeworks will be accepted.* The lowest homework score will be dropped. There will be 6 or 7 homework assignments. Some of the homework assignments may be time-consuming. You are encouraged to "discuss" the problem sets and study together in group, but when it comes to formulating/writing solutions you must work alone independently. Copying homework solutions from another student will be considered cheating. Instances of academic dishonesty will be dealt with harshly. (See the Honor Code below.)

As a courtesy to the grader, homework should be written neatly. Poorly written homework sets will not be graded. When writing algorithms be sure not only that your solution is correct, but also that it is easy for the grader to understand why your solution is correct. Part of your grade will be based not only on correctness, but also on the clarity, simplicity, and elegance of your solution.

Communication: Both the instructor's and TA's office hours will be as posted, or by appointment and email correspondence. A class mailing list will be set up to broadcast important messages related to the class. Please send mail to lin@cs.unc.edu to add yourself to the mailing list.
Lecture notes, homework problems sets, handouts and class announcements will also be posted on the course home page at the following URL:

`http://www.cs.unc.edu/~lin/COMP122/`

Make-up Course Work: In exceptional circumstances (serious illness, university business, a death in the family), an extension or a make-up exam may be granted. (The problems of student life, including the consequences of procrastination and commitments to other courses are not regarded as "exceptional circumstances".) However, all extensions or alternative arrangements must be approved by the instructor BEFORE the due date. In circumstances that merit special consideration, documentation is usually available to the student, and I feel most comfortable when a request for make-up work is accompanied by appropriate written material supporting such a request.

Courtesy: We will try to be courteous to you and we ask that you be courteous to others as well. Please do not read newspaper or other materials during class. If you must be late once or twice, take an aisle seat quitely; likewise if you must leave early. If this becomes habitual, you should drop the course. (This course will be offered in Fall 1999.) Most importantly, please remain quiet, except, of course, to ask questions or answer questions posed by the instructor/TA. Private discussion between students, even whispers, carry surprisingly well and are a real distraction to those seating near you and to the instructor/TA. Thank you.

How to Succeed in This Course:

1. Start early on ALL assignments. *DON'T* Procrastinate.
2. Complete all reading assignments before class.
3. Participate in class.
4. Think in class.
5. Review after each class.
6. Be formal and precise on all problems sets and exams.

Honor Code: The Honor Code and the Campus Code are in effect for this course. The following is adapted from a recent memo from Chancellor Hooker.

> The Honor Code prohibits lying, cheating, or stealing when these actions involve academic processes or University, students, or academic personnel acting in an official capacity. The Campus Code requires students to conduct themselves so as not to impair significantly the welfare or the educational opportunities of others in the University community. As a student at UNC-CH, you have accepted a commitment to the Honor Code and the Campus Code, and the principles of academic integrity, personal honesty, and responsibility on which they were founded more than 100 years ago.
>
> Academic dishonesty in any form is unacceptable, because it circumvents the purpose of the University. The instructor and teaching assistant have a responsibility to report any possible Honor Code violations to the Student Attorney General. Please join us in supporting the Honor Code by signing the Honor Pledge on all written work, and consult us if you are uncertain about your responsibilities within this specific course.

Topics Covered	Classes
Assume You Know:	
Chapter 2.2, 3, 5.1-5.3, 5.5, 11, 12.1-12.2	0
Administrivia	0.5
Chapter 1 -- Introduction to the course	0.5
Basic Analytical Tools	
Chapter 2 -- Asymptotic notation	0.5
Chapter 4 -- Recurrences	1.5
Chapter 6 -- Counting and Probability	2
Sorting & Ordering	
Chapter 7 -- Heapsort	1
Chapter 8 -- Quicksort	1
Handout on Other Sorting Methods	1
Chapter 9 -- Sorting in Linear Time	1
Chapter 10 -- Medians and Order Statistics	1
Chapter 12 -- Hash Tables	1
Balanced Trees	
Chapter 13 -- Binary Search Trees	0.5
Chapter 14 -- Red-Black Trees	0.5
Chapter 19 -- B-Trees	1
Advanced Design and Analysis Techniques	
Chapter 16 -- Dynamic Programming	2
Greedy Algorithms	
Chapter 17 -- Greedy Algorithms (will skip 17.4 and 17.5)	1
Chapter 24 -- Minimum Spanning Tree	1
Chapter 25 -- Single-Source Shortest Paths (25.2 on Dijkstra's)	1
Graph Algorithms	
Chapter 5.4 -- Graphs	0
Chapter 23 -- Elementary Graph Algorithms	2
Chapter 27 -- Maximum Flow (if time allows)	1
Selected Topics	
Chapter 36 -- NP-Completeness	1
Chapter 37 -- Approximation Algorithms	1
Others -- Applications, Geometric Algorithms, etc.	3