Course Description

Instructor: Prof. Richard Chang
Office: ITE 326
Office Hours: Tuesday & Thursday, 11:30am - 12:30pm
Telephone: 455-3093 (office), 455-3500 (department office)
E-mail: chang@umbc.edu

The TA's office hours will be announced at a later date.

Time and Place. Tuesday & Thursday 2:30pm – 3:45pm, ITE 231.

Textbook. *Introduction to Algorithms,* second edition, Cormen, Leiserson, Rivest and Stein. McGraw-Hill. ***Note:*** you must have the second edition of this textbook which was published in August 2001. The new textbook has a green cover.

Course Web Page. <http://www.csee.umbc.edu/~chang/cs441.f03>

Prerequisites. Students taking CMSC 441 should have mastered the material covered in the following courses: CMSC 201 & 202 (Computer Science I & II), CMSC 203 (Discrete Structures), CMSC 341 (Data Structures) and MATH 152 (Calculus and Analytic Geometry II). The material in Appendix B, Chapter 10 and Chapter 12 of the textbook (covering sets, elementary data structures and binary search trees) should be familiar. So should sorting algorithms (e.g., bubble sort, insertion sort, selection sort, merge sort, heap sort, quicksort). Some knowledge of probability and counting (Appendix C of the textbook) is also expected. In addition, proficiency in the implementation of the elementary data structures (e.g. stacks, queues, linked lists, heaps, balanced binary trees and graphs) in C/C++ or Java is assumed.

A 30-minute prerequisite test will be given on Thursday, September 4. This test will cover material that you should already know from CMSC 203 and CMSC 341. The test counts for 3% of your final grade.

Objectives. In this course students will

1. learn the quantitative methods used in the analysis of algorithms;
2. sharpen their problem solving skills through the design of algorithms; and
3. learn to write explanations for the correctness of algorithms and justifications for their performance.

A secondary goal of this course is to familiarize students with a range of fundamental algorithms.

Grading. Final grades will be based upon the prerequisite test (3%), 9 homework assignments (27% total), 2 writing assignments (20% total), the midterm exam (20%) and the final exam (30%). Grades are given for work done *during* the semester; incomplete grades will only be given for medical illness or other such dire circumstances.

The final letter grades are based on the standard formula:

$$0 \le F < 60, \quad 60 \le D < 70, \quad 70 \le C < 80, \quad 80 \le B < 90, \quad 90 \le A \le 100$$

Depending upon the distribution of grades in the class, there may be adjustments in the students' favor, but under no circumstances will the letter grades be lower than in the standard formula. Grades will not be "curved" in the sense that the percentages of A's, B's and C's are not fixed. As a guideline, a student receiving an "A" should be able to solve the homework problems with facility; design and analyze new algorithms in written exams; and demonstrate an understanding of the impact of theoretical analysis in practical programming settings.

Lecture and Homework Policy. Students are expected to attend all lectures and are responsible for all material covered in the lecture as well as those in the assigned reading. However, this subject cannot be learned simply by listening to the lectures and reading the book. In order to master the material, you must spend time outside the classroom, to think, to work out the homework and understand the solutions.

Assignments are due at the *beginning* of lecture. *Late homework will not be accepted — this is to allow for timely grading and discussion of the homework solutions*. Reasonable provisions will be made for students who are delayed by traffic, who are on travel, ... *Late homework will be rejected from students who have obviously been working on homework instead of attending lecture*. Partial credit will be given for serious attempts on the homework problems. So you should simply turn in whatever you have accomplished by the beginning of class. If you cannot attend lecture when homework is due, for some honorable reason, you must make arrangements to submit your homework directly to the instructor. Do not ask another student to submit your homework for you. This is to reduce the temptation to cheat (see below).

Working Together. Students are allowed, and even encouraged, to collaborate on homework problems. Collaborators and reference materials must be acknowledged at the top of each homework assignment. However, homework solutions must be written up *independently*. A student who is looking at someone else's solution or notes, whether in print or in electronic form, while writing up his or her own solution is considered to be cheating. Cases of academic dishonesty will be dealt with severely.

Writing assignments, on the other hand, must be accomplished through individual effort. Students may get help on grammar, sentence and paragraph construction from the Writing Center or from others who are not taking this course, but all technical work must be done individually. *Writing assignments that are found to have as much as a single sentence copied will receive a grade of zero. This penalty applies to both the copier and to the copied. So, do not allow another student to look at your writing assignment.*

Exams. The exams will be closed-book and closed-notes. The midterm exam will be given on Tuesday, October 21. The final exam will be comprehensive and cover the material from the entire course. The final exam will be given on Thursday, December 11, 1:00pm–3:00pm.

Course Syllabus

We will follow the textbook *Introduction to Algorithms*, second edition, by Cormen, Leiserson, Rivest and Stein. The following schedule outlines the material to be covered during the semester and specifies the corresponding sections of the textbook.

	Date	Topic	Reading	HW assigned	HW Due	Writing assigned	Writing Due
Th	08/28	Introduction, Proofs	1.1-3.2				
Tu	09/02	Greedy Algorithms	16.1-16.2			Draft	
Th	09/04	Summations, PREREQ TEST	A.1-A.2				
Tu	09/09	Recurrences	4.1-4.2	HW1			
Th	09/11	Master Theorem	4.3-4.4				
Tu	09/16	Heapsort	6.1-6.5	HW2	HW1		
Th	09/18	Quicksort	7.1-7.4				Draft
Tu	09/23	Lower bounds on Sorting	8.1-8.4	HW3	HW2		
Th	09/25	Linear-Time Selection	9.1-9.3				
Tu	09/30	Hash Tables	11.1-11.5	HW4	HW3		
Th	10/02	Dynamic Programming I	15.1-15.3			Revision	
Tu	10/07	Dynamic Programming II	15.4-15.5	HW5	HW4		
Th	10/09	Greedy Algorithms Revisited	16.3				
Tu	10/14	Dynamic Programming vs Greedy			HW5		
Th	10/16	Review					Revision
Tu	10/21	**Midterm Exam**					
Th	10/23	Basic Graph Algorithms I	22.1-22.4			Draft	
Tu	10/28	Basic Graph Algorithms II					
Th	10/30	Basic Graph Algorithms III		HW6			
Tu	11/04	Strongly Connected Components	22.5				
Th	11/06	Minimum Spanning Trees I	23.1-23.2	HW7	HW6		
Tu	11/11	Minimum Spanning Trees II					Draft
Th	11/13	Shortest Paths I	24.1-24.3	HW8	HW7		
Tu	11/18	Shortest Paths II	24.4-24.5				
Th	11/20	Shortest Paths III	25.1-25.3		HW8	Revision	
Tu	11/25	Maxium Flow I	26.1-26.3				
Th	11/27	*Thanksgiving break*					
Tu	12/02	Maximum Flow II		HW9			
Th	12/04	NP-completeness					Revision
Tu	12/09	Review			HW9		
Th	12/11	**Final Exam 1:00pm - 3:00pm**					