CSCI 445 SYLLABUS

COURSE NAME, NUMBER	CSCI 445: Analysis of Algorithms
SEMESTER, YEAR	Fall 2008
INSTRUCTOR	T. VanDrunen
OFFICE / TELEPHONE / EMAIL	Armerding 112 752-5692 Thomas.VanDrunen@wheaton.edu
OFFICE HOURS	MTuWThF 2:00–3:00 pm, Th 9:00–11:00 am
COURSE WEBSITE	`http://cslab.wheaton.edu/~tvandrun/cs445`

RESOURCES

Cormen, Thomas et al. *Introduction to Algorithms*, second edition, McGraw Hill / MIT Press, 2001.
Lewis and Papadimitriou. *Elements of the Theory of Computation*, second edition, Prentice Hall, 1998

COURSE DESCRIPTION

An introduction to algorithmic efficiency and to techniques for the design and analysis of efficient algorithms. General topics include a review of asymptotics, algorithm design techniques (such as divide-and-conquer, dynamic programming, and greedy algorithms), graph algorithms, *languages and automata*, and NP-completeness.

GOALS AND OBJECTIVES

1. Students will be able to analyze the time complexity of algorithms
   - Using worst- and average-case analysis.
   - Using recurrences and substitution.
   - Using the master method.
   - Using amortized analysis.
   - Using rigorous experimental methods
2. Students will be able to understand, adapt, and devise advanced algorithms
   - For classical problems such as sorting and searching.
   - For specialized sets of data.
   - Using advanced and specialized data structures.
   - Using dynamic programming and greedy strategies.
3. Students will be able to identify and categorize intractable problems
   - In articulating the nature of NP-completeness.
   - In proving problems to be NP-hard and NP-complete.
   - In finding approximate solutions to NP-complete problems.

ASSESSMENT PROCEDURES

1. Problem sets will exercise students' abilities to compose algorithms, analyze their complexity, and prove facts about them.
2. Two large programming projects will teach students to connect the theoretical approach to algorithms used in class to the real task of programming and of experimental testing of software.
3. The midterm and final exam will evaluate students' mastery of the comprehensive material.

Grading:

	weight
Homework	30
Big project 1	15
Big project 2	15
Midterm	20
Final	20

SPECIAL EXPECTATIONS

Academic Integrity

Homework: Collaboration among students in the class is permitted on most assignments. Many problems you will be assigned are discussed or solved in the computer science literature or on the Internet. Use of these resources for graded assignments is discouraged, and if they are ever used, full citation must be given, just as in a research paper. Use of any resources that specifically serve as solutions to exercises in Cormen et al is not permitted.

Projects: Work on projects must be done independently. Students should not receive help (except for minor debugging advice) from anyone, inside or outside the course, except the instructor. Any ideas (such as algorithms, the design of data structures, an analysis) received from outside resources must be cited.

Examinations: Any take-home portion of the midterm or final must be completed independently, without any aide from any person inside or outside the class. Unless stated otherwise, assume take-home portions of the examinations are closed-book, closed-notes and that the use of other resources is not permitted.

Attendance

Students are expected to attend all class periods. It is courtesy to inform the instructor when a class must be missed.

Late Assignments

Late projects and graded homework will not be accepted. If an assignment or project is incomplete by the deadline, turn in what you have for partial credit.

Special needs

Whenever possible, classroom activities and testing procedures will be adjusted to respond to requests for accommodation by students with disabilities who have documented their situation with the Registrar and who have arranged to have the documentation forwarded to the course instructor. Computer Science students who need special adjustments made to computer hardware or software in order to facilitate their participation must also document their needs with the Registrar in advance before any accommodation will be attempted.

Projects. There will be two big projects during the course of the semester, one in each quad. Both will involve programming, experimentation, and analysis. Both will require writing a report.

Textbook. Students are expected to purchase Cormen et al. To reduce the cost to students, the computer science department has purchased four copies of Lewis and Papadimitriou, which students may borrow.

Class format. This course will not follow a typical lecture format. To prepare for a class period, students will read a portion of the textbook and work on a few exercises based on the reading. Class will be spent clarifying and discussing points in the reading and working out problems and exercises.

Topics. The primary textbook for this course ("CLRS") is the standard book for advanced undergraduate and introductory graduate study in the field of algorithms, and we will follow it closely. We will cover most of Parts I, II, IV, and VI; time an student interest will govern how much of the selected topics in Part VII we can cover; we will not cover Parts III and V as that material is adequately addressed in CSCI 345. Broadly, the topics of this course are divided into five modules:

I. Elementary analysis techniques (CLRS 1–4, 7, 8, 27). 3-5 weeks.

> *These techniques are called "elementary" in comparison to some of the advanced techniques we use later, but some are quite sophisticated in their own right. We will apply these techniques to sorting, the standard algorithmic example problem.*

II. Advanced algorithm and analysis techniques (CLRS 15–17). 2–3 weeks.

> *Dynamic programming, greedy algorithms, and amortized analysis.*

III. Graph algorithms (CLRS 22, 24–26). 2–3 weeks.

> *Basic graph algorithms (BFS, DFS, etc) will be reviewed as necessary. Our focus will be on single-source shortest paths and all-pairs shortest paths algorithms. We will also study flow networks.*

IV. Miscellaneous topics (CLRS 21, 27–33). 1–3 weeks.

> *With what time we have before we begin the module on theory, we will choose from the following topics:*
>
> *Dynamic disjoint sets (CLRS 21)*
> *Linear programming (CLRS 29)*
> *String-matching (CLRS 32)*
> *Computational geometry (CLRS 33)*
>
> *These, at least, are what I suggest. We could cover any chapter in CLRS part VII if there is sufficient student interest.*

V. Theory and NP-Completeness. 4 weeks.

> *Most of this will come from our secondary textbook ("LP"). Automata and languages (LP 2 and 3) will be introduced and reviewed as necessary. Our focus will be on Turing machines (LP 4), computational complexity (LP 6), and NP-completeness (LP 7). We will also look at an algorithmic approach to understanding and dealing with NP-completeness (CLRS 34 and 35) for comparison. To ensure sufficient time before the end of the semester, this module will begin on Nov 10.*

See the course website for the current plan for specific topics and their order. This schedule is subject to change.

Examinations. The midterm is Friday, Oct 17. That is the week of fall break. The final exam is on Tuesday, Dec 16, 8:00 am. I do not allow students to reschedule examinations because of travel, so make your plans accordingly. The final will be mostly non-cumulative.