COMP 460: Algorithms & Complexity (Section 001)
Fall 2004 Course Information & Syllabus

Instructor: R. I. Greenberg
Computer Science Department
Loyola University
Water Tower Campus, LT-512E
820 N. Michigan Ave.
Chicago, Illinois 60611-2147

Phone: (312)915-7981 **Email:** rig@cs.luc.edu **Home page:** http://www.cs.luc.edu/~rig

Lectures: TuTh 1:00–2:15 pm in 25EP-0206.
Sometimes lecture notes or a summary will be available on the web. Other than that, if you have to miss a class, get notes from another student; mine are typically pieced together from more than one place with a lot of metacomments, which makes it hard for anybody but me to follow them. Also get copies of any missed handouts (available on the web site). The handouts are numbered sequentially, starting with handout 0. On handout 0, you need to fill in some information and return it to me promptly so you can be on the email list and get access to the web site for the course.

Office Hours: In Lewis Towers 512E: 10:30–11:30 and 2:15–3:00 on Tuesday and Thursday.
These are the guaranteed times to find me except as announced in advance. You should also be able to find me at lots of other times; feel free to look for me or make appointments.

Course Objectives: This course will focus both on presenting general techniques for designing correct and efficient algorithms, as well as on formal methods for proving the correctness and analyzing the complexity of such algorithms. The course assumes some prior mathematical and algorithmic background and is intended to broaden and deepen that knowledge. Also included will be an introduction to the theory of NP-completeness, whereby certain computation problems can be classified as being difficult in a formal sense.

Prerequisites: Introduction to the Design and Analysis of Algorithms (COMP 363) or Foundations of Computer Science (COMP 388 Section 031 in Fall 2001 or Section 033 in Spring 2002).

Required Text: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* McGraw-Hill, second edition, 2001.

Course Requirements: There will be several homework assignments, two midterm exams, and a final. The weightings within the semester grade will be: Homework 20%, Exam I 20%, Exam II 25%, and Final exam 35%.

Homework: Only homework turned in by the due date is guaranteed to be graded. Any special circumstances that cause difficulty in meeting the deadlines should be brought to the attention of the instructor in advance. Homework must be handed in at the beginning of class, since solutions may be handed out in the same class on occasion. Homework turned in to my mailbox will generally not be graded, since I do not check the box continually and cannot generally verify that homework was turned in before solutions were distributed or discussed in class. If you cannot turn in homework in person, you should put it under the door of my office.

Exams: The midterm exams, tentatively scheduled for week 6 and week 13, are 75 minutes long. The final exam is scheduled for 11:00am–1:00pm on Thursday, December 16.

Collaboration: *No* collaboration is permitted on exams. *Collaboration* on homework is acceptable, but *copying* is not! (Safeguard your files and printouts.) You may discuss solution techniques with other students, but you must write up your solutions independently. If you obtain a solution through research, e.g., in the library, credit your source and write up the solution in your own words.

Tentative Course Outline and Approximate Schedule:

Recommended readings from the text are shown on a weekly basis. (When selected sections or subsections are listed, it is assumed that you will include the introduction of the corresponding chapter or section.)

(Topics not mentioned explicitly in the syllabus that you will be assumed to be mostly familiar with are Chapter 10 on Elementary Data Structures, the Appendices (excepting C.5) on summations, sets, counting, and probability, and Sections 12.1–3 on binary search trees.)

1. (8/31) Administrivia, introduction. Review of growth of functions and recurrences. Insertion sort and Merge sort. Chapters 1–4.
   Heapsort. Chapter 6.

2. (9/7) Quicksort. Chapter 7. Useful background is also included in Chapter 5 through page 100.
   Sorting in linear time. Chapter 8.

3. (9/14) Order statistics. Chapter 9.

4. (9/21) Hashing. Sections 11.1–4 (especially after Section 11.3.2; most of the earlier material should be review).
   Red-black trees. Chapter 13.

5. (9/28) Augmenting Data Structures. Chapter 14.

6. (10/5) Dynamic Programming. Sections 15.1–3.
   Exam I on material through *week of* 9/28.

7. (10/14) Greedy Algorithms. Sections 16.1–3.

8. (10/19) Amortized Analysis. Chapter 17.
   Union-find. Sections 21.1–3.

9. (10/26) Review of elementary graph algorithms. Strongly connected components Chapter 22.
   Single-source shortest paths. Sections 24.1–3

10. (11/2) All-pairs shortest paths. Chapter 25.
    Maximum flow. Sections 26.1–3.

11. (11/9) Matrix operations. Sections 28.1–4.
    Linear programming. Chapter 29

12. (11/16) Polynomials and the FFT. Chapter 30.
    Elementary number-theoretic algorithms. Sections 31.2, 4, and 7. (Background reference: Sections 31.1, 3, and 6.)

13. (11/23) Exam II on material from week of 10/5 through week of 11/9.

14. (11/30) String matching. Sections 32.1–3.

15. (12/7) NP-completeness background: polynomial time, decision problems, determinism versus nondeterminism. Section 34.1. NP-completeness: solution versus verification, reducibility and definition of NP-completeness. Sections 34.2–3.
    Proving problems NP-complete, and examples. Sections 34.3–5.