COMP 363: Design and Analysis of Computer Algorithms (Section 001)
Spring 2005 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 4:15–5:45 pm in 25EP-0105.
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: 1:00–4:00pm on Tuesday and Thursday.
These are the guaranteed times to find me except as announced in advance. You should also feel free to look for me at other times or make appointments.

Course Objectives: Students should gain basic skills in designing and implementing efficient and effective computer programs. We will begin by developing models and mathematical tools for measuring the efficiency of algorithms. Then students will be introduced to a variety of useful data structures and to algorithms for a variety of fundamental problems. Finally, the course will provide an introduction to classification of computational problems into different complexity classes. The course may include a small amount of actual running of programs in addition to theoretical analysis.

Prerequisites: The formal prerequisites are COMP 271 and COMP 211 (which require COMP 170 and MATH 118). The actual desired topical backgound includes: programming in a general-purpose language including capabilities for recursion (e.g., Java or C), induction, basic calculus, basic probability, sets, permutations and combinations. Familiarity with graphs and trees also provides a helpful head start. Except for the programming, review material on most of these topics is included in the textbook. Also, there is a bug in the evolving prerequisite structure in that one should have MATH 131 or MATH 161 before COMP 363. (This used to be a prereq for COMP 211.) The desired skills from MATH 131 or MATH 161 are to be able to compute limits using L'Hospital's Rule and to be able to find minima and maxima of a function by differentiating.

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 4:15–6:15pm on Thursday, May 12.

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.)

Also included are some sections that may be consulted as needed for background on topics such as mathematical notations, probability, graph theory, etc. I've tried to flag the first point at which such sections may be useful.

1. (1/18) Administrivia, introduction, analysis of algorithms. The divide-and-conquer paradigm. Chapters 1 and 2.
2. (1/25) Growth of functions. Sections 3.1–2. Recurrences. 4.1–2. (Background reference: Section 3.2, Appendix A.)
3. (2/1) Recurrences continued. Section 4.3. Heapsort. Chapter 6. (Background reference: Section B.5.3.)
4. (2/8) Quicksort. Chapter 7. (Background reference: Section C.2: "Axioms of probability", and "Discrete probability distributions", Section C.3.) Sorting: A lower bound in the comparison model, and sorting in linear time. Chapters 8 (Background reference: Section C.4: "The binomial distribution" through page 1115.
5. (2/15) Order Statistics. Chapters 9. Stacks, queues, linked lists. Sections 10.1–2.
6. (2/22) Review for exam 1 and/or catchup on lecture material. Exam I on sections covered from parts I and II of book.
7. (3/1) Hashing. Sections 11.1–4.
8. (3/15) Binary search trees. Sections 12.1–3. Red-black trees. Sections 13.1–4.
9. (3/22) Dynamic Programming. Sections 15.1–3
10. (3/29) Dynamic Programming continued. Sections 15.3–4. Greedy Algorithms. Sections 16.1–2.
11. (4/5) Representation of graphs. Breadth-first search. Sections 22.1–4. (Background reference: Sections B.4–5.) Depth-first search & applications. Sections 22.3–4.
12. (4/12) Minimum spanning tree algorithms. Sections 23.1–2. Single-source shortest paths, Bellman-Ford, and dags. Sections 24.1–2.
13. (4/19) Dijkstra's algorithm. Section 24.3. (We will use some results from section 21.3.) Exam II on sections covered from parts III and IV of book plus Chapters 22–23.
14. (4/26) All-pairs shortest paths. Chapter 25. 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.
15. (5/3) Proving problems NP-complete, and examples. Sections 34.3–5. Review or additional enrichment material or overflow from prior lectures.