

CS 373: Combinatorial Algorithms

University of Illinois, Urbana-Champaign

Instructor: Jeff Erickson

Teaching Assistants:

Spring 1999: Mitch Harris and Shripad Thite

Summer 1999 (IMCS): Mitch Harris

Summer 2000 (IMCS): Mitch Harris

Fall 2000: Chris Neihengen, Ekta Manaktala, and Nick Hurlburt

Spring 2001: Brian Ensink, Chris Neihengen, and Nick Hurlburt

© Copyright 1999, 2000, 2001 Jeff Erickson.

This work may be freely copied and distributed.

It may not be sold for more than the actual cost of reproduction.

For the most recent edition, see <http://www.uiuc.edu/~jeffe/teaching/373/>.

For junior faculty, it may be a choice between a book and tenure.

— George A. Bekey, “The Assistant Professor’s Guide to the Galaxy” (1993)

I’m writing a book. I’ve got the page numbers done.

— Stephen Wright

About These Notes

This course packet includes lecture notes, homework questions, and exam questions from the course ‘CS 373: Combinatorial Algorithms’, which I taught at the University of Illinois in Spring 1999, Fall 2000, and Spring 2001. Lecture notes and videotapes lectures were also used during Summer 1999, Summer 2000, and Summer 2001, as part of the UIUC computer science department’s Internet Master of Computer Science program. Mitch Harris, the teaching assistant for the first two summer courses, used different homework and exam problems, which are also included here.

The recurrences handout is based on samizdat (based in turn on a paper of George Lueker) from an earlier semester taught by Ed Reingold. I wrote the lecture notes mostly in Spring 1999; I revised them and added a few new notes in Fall 2000 and Spring 2001. Except for the infamous Homework Zero, which is entirely my doing, homework and exam problems and their solutions were written mostly by the teaching assistants: Brian Ensink, Chris Neihengen, Ekta Manaktala, Mitch Harris, Nick Hurlburt, and Shripad Thite. Lecture notes were posted to the course web site a few days (on average) after each lecture. Homeworks, exams, and solutions were also distributed over the web. I have deliberately excluded solutions from this course packet, but I will gladly make them available to interested instructors.

The lecture notes, homeworks, and exams draw heavily on the following sources, all of which I can recommend as good references.

- Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974. (This was the textbook for the algorithms classes I took as an undergrad at Rice and as a masters student at UC Irvine.)
- Sara Baase and Allen Van Gelder. *Computer Algorithms: Introduction to Design and Analysis*. Addison-Wesley, 2000.
- Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 1997. (This is the required textbook in my computational geometry course.)
- Thomas Cormen, Charles Leiserson, and Ron Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990. (This is the required textbook for CS 373, although I never actually use it in class. Students use it as a educated second opinion. I also used this book as a teaching assistant at Berkeley.)
- Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Molecular Biology*. Cambridge University Press, 1997.
- Udi Manber. *Introduction to Algorithms: A Creative Approach*. Addison-Wesley, 1989. (I used this textbook as a teaching assistant at Berkeley.)

- Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- Ian Parberry. *Problems on Algorithms*. Prentice-Hall, 1995. (This was a recommended textbook for CS 373, primarily for students who needed to strengthen their prerequisite knowledge.)
- Robert Sedgewick. *Algorithms*. Addison-Wesley, 1988. (This book and its sequels have by far the best algorithm *illustrations* anywhere.)
- Robert Endre Tarjan. *Data Structures and Network Algorithms*. SIAM, 1983.
- Various journal and conference papers (cited in the notes).
- Class notes from my own algorithms classes at Berkeley, especially those taught by Dick Karp and Raimund Seidel.

Naturally, everything here owes a great debt to the people who taught me this algorithm stuff in the first place: Bob Bixby, Manuel Blum, David Eppstein, Dan Hirshberg, Dick Karp, Mike Luby, George Lueker, Michael Perlman, Abhiram Ranade, Ed Reingold, and Raimund Seidel. I stole the overall course structure from Herbert Edelsbrunner.

We did the best we could, but I'm sure there are still plenty of mistakes, errors, bugs, gaffes, omissions, snafus, kludges, typos, mathos, grammaros, thinkos, brain farts, nonsense, garbage, cruft, junk, and outright lies, all of which are entirely Steve Skiena's fault. I plan to revise and update these notes every time I teach CS 373, so if you find a bug, please let me know. (Steve is unlikely to care.) Of course, any other feedback is also welcome.

I will award extra credit points to the first CS 373 student to post an explanation and correction of any error in the lecture notes to the course newsgroup (uiuc.class.cs373). Obviously, the number of extra credit points depends on the severity of the error and the quality of the correction.

The drawings that appear in the hardcopy version of this course packet were contributed by students. The pseudonymous "Dr. Orcface" note was taped to my office door when I fell a week behind in the lecture notes in Spring 1999. "Johnny's" multi-colored crayon homework was found under the TA office door among the other Fall 2000 Homework 1 submissions. The portrait of the Partial Credit Fairy is from Joel Meyer's Fall 2000 final exam. The drawing of Cerebus the Aardvark is by Timothy Hammack, from his Spring 1999 final exam 'cheat sheet'.

Enjoy!

— Jeff

It is traditional for the author to magnanimously accept the blame for whatever deficiencies remain. I don't. Any errors, deficiencies, or problems in this book are somebody else's fault, but I would appreciate knowing about them so as to determine who is to blame.

— Steven S. Skiena, *The Algorithm Design Manual*, Springer-Verlag, 1997, page x.