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
- **Summer 2001 (I2CS):** Asha Seetharam and Dan Bullok
- **Fall 2002:** Erin Wolf, Gio Kao, Kevin Small, Michael Bond, Rishi Talreja, Rob McCann, and Yasutaka Furakawa

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, Spring 2001, and Fall 2002. Lecture notes and videotapes lectures were also used during Summer 1999, Summer 2000, Summer 2001, and Fall 2002 as part of the UIUC computer science department's Illinois Internet Computer Science (I2CS) program.

The recurrences handout is based on samizdat, probably written by Ari Trachtenberg, based on a paper by George Lueker, from an earlier semester taught by Ed Reingold. I wrote most of the lecture notes in Spring 1999; I revised them and added a few new notes in each following semester. 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: Asha Seetharam, Brian Ensink, Chris Neihengen, Dan Bullok, Ekta Manaktala, Erin Wolf, Gio Kao, Kevin Small, Michael Bond, Mitch Harris, Nick Hurlburt, Rishi Talreja, Rob McCann, Shripad Thite, and Yasu Furakawa. 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.

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, Ron Rivest, and Cliff Stein. *Introduction to Algorithms*, second edition. MIT Press/McGraw-Hill, 2000. (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 used the first edition of 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.
- Michael T. Goodrich and Roberto Tamassia. *Algorithm Design: Foundations, Analysis, and Internet Examples*. John Wiley & Sons, 2002.
- 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 early versions of CS 373, primarily for students who needed to strengthen their prerequisite knowledge. This book is out of print, but can be downloaded as karmaware from http://hercule.csci.unt.edu/~ian/books/poa.html .)
- 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.
- Class notes from my own algorithms classes at Berkeley, especially those taught by Dick Karp and Raimund Seidel.
- Various journal and conference papers (cited in the notes).
- Google.

Naturally, everything here owes a great debt to the people who taught me this algorithm stuff in the first place: Abhiram Ranade, Bob Bixby, David Eppstein, Dan Hirshberg, Dick Karp, Ed Reingold, George Lueker, Manuel Blum, Mike Luby, Michael Perlman, and Raimund Seidel. I've also been helped immensely by many discussions with colleagues at UIUC—Ed Reingold, Edgar Raoms, Herbert Edelsbrunner, Jason Zych, Lenny Pitt, Mahesh Viswanathan, Shang-Hua Teng, Steve LaValle, and especially Sariel Har-Peled—as well as voluminous feedback from the students and teaching assistants. I stole the overall course structure (and the idea to write up my own lecture notes) 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 revise and update these notes every time I teach the course, so please let me know if you find a bug. (Steve is unlikely to care.)

When I'm teaching CS 373, I award extra credit points to the first 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. If I'm not teaching the course, encourage your instructor to set up a similar extra-credit scheme, and forward the bug reports to ~~Steve~~ me!

Of course, any other feedback is also welcome!

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.