Computer Science 511
Design and Analysis of Algorithms
Fall 2011
www.cs.iastate.edu/∼cs511

General Information

Description. A study of basic algorithm design and analysis techniques. Emphasis is on learning to formulate algorithmic problems and on developing problem-solving skills

Prerequisites. CS 311 (undergraduate design and analysis of algorithms) or equivalent. A strong background in programming and discrete mathematics is essential.

Time and Place. MWF 11:00–11:50 a.m., Science 0102

Instructor. David Fernández-Baca, fernande@iastate.edu, 294-2168, 111 Atanasoff. Office hours are by appointment, but students may drop in whenever the instructor's door is open.

Teaching Assistants. TBA.

References. Our textbook is

> Jon Kleinberg and Éva Tardos, *Algorithm Design*, Addison-Wesley, 2006.

The following is a good introduction to algorithms.

> Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani, *Algorithms*, McGraw-Hill 2007. A near-final version of this text is available at http://www.cs.berkeley.edu/∼vazirani/algorithms.html

The next reference offers near-encyclopedic coverage of the field.

> Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms* (3rd edition), MIT Press, 2009.

For additional material on specific topics, you might want to consult the following books.

> Robert J. Vanderbei. *Linear Programming: Foundations and Extensions* (3rd edition). Springer, 2010.

Rolf Niedermeier, *Invitation to Fixed-Parameter Algorithms*, volume 31 of Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, Oxford, 2006. An early version of this book can be found at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.9618

David P. Williamson and David B. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, April 2011. An electronic version of the book can be downloaded from http://www.designofapproxalgs.com/download.php

Topics

All chapter references below are to the Kleinberg-Tardos text.

1. Flows and cuts (3 weeks; Chapter 7)
2. Linear programming[1] (2 weeks)
3. NP-completeness (2 weeks; Chapter 8)
4. Fixed-parameter algorithms (3 weeks; Chapter 10)
5. Approximation algorithms (3 weeks; Chapter 11)
6. Randomized algorithms (2 weeks; Chapter 13)

Grading

Final grades will be based on scores on homework assignments, 2 midterm exams, and a final. The weights of these are as follows:

Homework	10%
Exam 1	30%
Exam 2	30%
Final	30%

Midterm exams will be 2-hours long and held in the evening. Exams 1 and 2 are tentatively scheduled for the weeks of September 26th and October 31st, respectively. The Final is tentatively scheduled for Tuesday December 13, 9:45-11:45 a.m.

[1] The instructor will provide supplementary material on this topic.

Policy on Academic Honesty[2]

Students enrolled in Computer Science courses at ISU are expected to maintain the highest standards of academic integrity. Suspected cases of academic misconduct will be pursued fully in accordance with university policies. Here we provide information to help students avoid unintentional misconduct. For more details, students are strongly urged to consult ISU's policy on academic dishonesty[3].

Problem Sets. The primary purpose of assignments is to clarify and enhance the understanding of the concepts covered in the lectures. It is expected that students have independently arrived at solutions that they turn in for problem sets. Nevertheless, past experience has shown that the learning experience is aided by interaction among classmates. Thus, discussion of general concepts and questions concerning the homework assignments among students is encouraged.

Examples of *allowed* collaboration are the following:

- Discussing the material presented in class or included in the assigned readings that is needed for solving assigned problems.
- Assisting other students in understanding the statement of the problem (e.g., by translating some English phrases unfamiliar to that student).

The following are examples of activities that are *prohibited*:

- Sharing solutions or fragments of solutions (via email, whiteboard, handwritten or printed copies, etc.).
- Posting solutions or fragments of solutions in a location that is accessible to others.
- Using solutions or fragments of solutions provided by other students (including students who had taken the course in the past).
- Using solutions or solution fragments obtained on the Internet or from solution manuals for text books.
- Using material from text books, reference books, or research articles without properly acknowledging and citing the source.

Exams. The primary purpose of exams is to test the understanding of the concepts covered in class. Exam solutions are expected to be entirely the work of each student. The following is a non-exhaustive list of activities that amount to *cheating* on an test.

- Copying someone else's solutions.

[2]This section is an abridged and adapted from a statement written by Professor Vasant Honavar for Computer Science 572. Thanks to Professor Honavar for allowing us to use this material.

[3]http://www.public.iastate.edu/~catalog/2005-07/geninfo/dishonesty.html

- Using notes or reference materials (unless instructed otherwise).
- Altering an exam for re-grading.
- Getting an advance copy of the examination.
- Having someone else write the exam.

Students with Disabilities

Iowa State University complies with the American with Disabilities Act and Section 504 of the Rehabilitation Act. Any student who may require an accommodation under such provisions should contact the instructor as soon as possible. No retroactive accommodations will be provided in this class.