CMSC 858S: Randomized Algorithms

Fall 2001

Handout 1: Basic course information

Instructor: Aravind Srinivasan - Office: AVW 3227. Office phone: 405–2695.
E-mail: srin@cs.umd.edu.
Office hours: Tuesday 2:00 - 3:00PM and Thursday 2:00 - 3:00PM.

Class Time: Fri 10-1, Room: CLB 0109

Class Webpage: *http://www.cs.umd.edu/~srin/fall01/fall01-home.html*

Course Overview: The powerful role played by randomness in computation has been among the central discoveries in the foundations of computer science over the last three decades. Randomized algorithms are algorithms that make random choices as they proceed, and have had a fundamental impact on several areas of computer science (e.g., distributed algorithms, cryptography, resource allocation, approximation algorithms). This course is an introduction to randomized algorithms. It aims to teach rigorous methods to design and analyze such algorithms, and to present some of the key applications of randomization in diverse fields of computer science such as distributed computing, resource allocation, scheduling, and packet routing.

We will cover basic paradigms, techniques to analyze randomized algorithms, as well as applications, in an intermingled fashion. Any such course can only cover a part of this field, and the contents of this course are somewhat biased according to the instructor's taste.

Text: *Randomized Algorithms* by R. Motwani and P. Raghavan, Cambridge University Press, 1995. We will distribute notes/papers for additional material covered. An excellent supplementary textbook is *The Probabilistic Method, Second Edition* by N. Alon and J. H. Spencer, Wiley, 2000. Brief notes will be given for material covered from this book that is not present in the Motwani-Raghavan book.

Prerequisites: Undergraduate algorithms, an interest in algorithms and their applications, and a basic exposure to the notion of NP-completeness (an in-depth understanding of computational complexity is not required). The mathematical background required is knowledge of elementary probability (mostly high-school-level), and the analysis techniques involved in an undergraduate algorithms course.

Coursework: The coursework will consist of a few homework assignments, a project, and two exams (one midterm and a comprehensive final). Homework problems will be mathematically oriented. The project will likely include a small programming assignment; students will get a few weeks to complete the project. Details of the project will be given within a few weeks.

The midterm will be in class in October (date will be posted on the webpage). Homeworks are to be turned in at the start of class on the due date. Since homework solutions will be handed out on the day the homework is due, NO LATE HOMEWORKS WILL BE ACCEPTED. (Hand in whatever you have finished. You are also welcome to turn in homeworks *before* the due date if you cannot come to class on the due date.) If you cannot come to class for some reason, please mail the homework to the instructor (should be postmarked a day **before** the due date).

All homeworks are to be done independently, with no help from your fellow students. If you find a solution from the web, a book, etc., please write down the solution in your own words, and include a reference to where you found the solutions. If you have questions, please talk to the instructor. Assignments are to be written up neatly; badly written assignments will not be graded. Please staple your homework. It is your responsibility to make sure that you pick up all homeworks and handouts. All course information and handouts will be available on the webpage.

Grading: Final grades will be based on homework assignments, the project, the midterm exam, and the comprehensive final exam. The relative weights of these will be 20% for the homework total, 10% for the project, 30% for the midterm, and 40% for the final exam.

Syllabus: The topics and order listed below are tentative and subject to change.

1. Basic probability: the power of the linearity of expectation, conditional probability, truncated inclusion-exclusion, some common distributions;
2. Paradigms underlying randomized algorithms, including: abundance of witnesses, symmetry-breaking, random sampling, and the probabilistic method;
3. Randomized complexity classes;
4. Large-deviation bounds (showing that certain types of random variables are typically very close to their expected value–such bounds are of much use in analyzing randomized algorithms);
5. The method of conditional probabilities (converting certain classes of randomized algorithms into efficient deterministic algorithms);
6. The Lovász Local Lemma (a powerful tool to prove the existence of some very low-probability events, with interesting applications, e.g., to packet routing in networks);
7. Randomized approximation algorithms;
8. The FKG and Janson inequalities, and the Poisson paradigm (with applications, e.g., to Steiner problems in networks);
9. Random graphs and expanders;
10. Applications: load-balancing, distributed contention resolution, low-congestion routing in networks, network design etc., and
11. A brief look at pseudorandomness, number-theoretic algorithms, quantum computing etc., if time permits.