comp363 Handout #1

Design and Analysis of Computer Algorithms

Michael Goldwasser

Loyola University Chicago Tuesday, 14 January 2003

Contents

1 Overview

Design and Analysis of Computer Algorithms is an undergraduate level course on the design and analysis of algorithms. The goal of the course is to convey that the eventual efficiency of a computer program for solving a problem can be most affected by high-level design decisions which are made before even the first line of code is written. Throughout the course, we will focus on presenting general techniques for designing efficient algorithms, as well as on classical methods for comparing the performance of different algorithms by formally analyzing their complexity (e.g. use of time, use of space, use of processors, quality of the output). We will apply these techniques to specific problems involving sequences, sets, graphs, as well as geometrical and numerical data. We will also ask the question of whether or not there exist problems for which there are no efficient techniques. This will lead us to the notion of NP-Completeness, and we will see examples of many such problems which are believed to be impossible to solve efficiently. For several of these problems, however, we will show that there are efficient algorithms which are able to find a solution which may not be the best, but which has cost provably close to the optimal.

1.1 The Instructor

instructor:	**Prof. Michael Goldwasser**
e-mail:	`mhg@cs.luc.edu`
web:	`http://www.cs.luc.edu/~mhg/`
office:	Damen Hall 319
telephone:	(773) 508-2883
office hours:	Tuesdays 5:00–5:45pm Thursdays 10:30–11:30am or by appointment

1.2 The Lectures

The material will be presented in one weekly lecture, meeting Tuesdays from 6:00–8:40pm, in room 608 of Mundelein Skyscraper (SKY). Class participation is most welcome.

1.3 Textbooks

The required textbook for this course is:

Title:	*Introduction to Algorithms, Second Edition*
Authors:	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein
Copyright Date:	2001.
ISBN:	0-07-013151-1, McGraw-Hill (internationally: 0-262-03293-7, MIT Press)
Web Sites:	`mitpress.mit.edu/algorithms` (MIT Press) `introtoalgorithms.com` (McGraw Hill)

The text should be available both at the Barnes and Noble campus bookstore, as well as Beck's Bookstore, or various online book vendors. A list of other useful reference books is included in Section 5.

1.4 Prerequisites

Students are assumed to enter this course with equivalent background in each of the following areas:

- *Discrete Structures (Comp 211).* A solid background in formal mathematics will prove invaluable for this course, in that we will argue rigorously about the correctness and performance of almost every algorithm and data structure we introduce. The ability to write clear and formal proofs will be relied upon heavily throughout the course.

- *Structured Programming & Data Structures (Comp 271).* Many of our algorithms will need to use more basic data structures as building blocks, and although we will try when possible to review the earlier material, we expect that students will have seen many of these concepts previously.

2 COMP 363 Web Page: `http://www.cs.luc.edu/~mhg/comp363`

We will attempt to keep all information related to the course online. This includes a schedule of lectures and readings, supplemental notes, assignments and due dates, as well as most solution sets.

The web page contains some information (e.g., supplemental notes, solutions, individual grades) which is more sensitive and therefore which will be available to students in the class only after they have identified themselves properly. To gain access to these parts of the web page, a student must first complete an online questionnaire, creating a unique identity and password.

3 Graded Work

3.1 Course Grades

The components of an overall course "percentage" will be calculated based on the following assigned work:

- **Homework (40%)** We expect there to be a total of six assignments during the course. We will ignore your lowest of the six grades, and the remaining five assignments will contribute to this portion of the grade. The collaboration policy and the late policy is discussed later in this document.

  The assignments will generally be of the "pen and paper" variety, due at the beginning of a class meeting. **They will be challenging.** You will have at least two weeks per assignment and we strongly suggest that you utilize the entire period in developing your solutions.

  The policies on collaboration, academic honesty and late assignments are outlined later in this handout.

- **Exams (60%)**
  - First Exam (15%), Tuesday, 18 February 2003
  - Second Exam (15%), Tuesday, 1 April 2003
  - Final Exam (30%), Tuesday, 6 May 2003

Details about the style and content of these exams will be provided as these dates approach.

A handout will be distributed immediately after the mid-semester break, defining precisely what percentages will be required for which letter grades.

3.2 Collaboration Policy

Each homework problem throughout the course will be designated with one of the two instructions:

- **"Work entirely on your own."**

  You should not discuss such problems with anyone other than the instructor, nor should you search for direct or indirect assistance from any outside references.

- **"You may discuss ideas with other students."**

  Even in this situation, students must not search for direct or indirect assistance from any outside references. Furthermore, **the write-up of your solutions must still be done individually, in entirety.**

  Our philosophy is that there are two distinct stages in working on each problem. First is reaching a point where you understand how to solve the problem, and second is formally writing up your solution in a clear way for a reader to understand.

  For problems where we allow collaboration, you are only allowed to collaborate on the first of these stages. Discussions with your peers are very helpful in understanding new material, and working through the puzzle of a new problem. You may therefore feel free to discuss with classmates general ideas, approaches, examples or stumbling blocks while trying to understand a homework problem.

  However, when it comes time to write up your solutions, you may not discuss this in any way with others, nor may you use anyone else's written solution as a guide. Our goal is that you will eventually be able to understand the problem so clearly that you are then able to lock yourself in a room with a blank piece of paper and produce a clear written explanation of your solution.

  When you do collaborate with classmates, you must work in groups of at most four students, and must write the names of all collaborators for that problem at the beginning of your solution.

3.3 Academic Honesty

Students are expected to have read the statement on academic integrity available on pages 12–13 of Loyola's "Undergraduate Studies" catalog. In addition to this statement, we wish to emphasize issues most relevant to this course.

When it comes to learning and understanding the general material covered in class, you may certainly use other references and you may have discussions with other students in this class or other people from outside of this class.

For written homework assignments, you must strictly follow the guidelines given in Section 3.2. For either designation of problems, you must not use or search for direct or indirect solutions using any outside references, including but not limited to:

- other texts or books
- any online information other than referenced in the course web page
- past students, whether from this school or other schools

Any violations of the general Loyola policy or the policies outlined in this handout will be dealt with severely. Penalties will apply as well to a student who is aiding another student. Any such violations will result in a minimum penalty of a zero on the given assignment which cannot be dropped, and severe or repeated violations will result in an immediate failing grade in the course. Furthermore all incidents will be reported in writing to both the department and the appropriate dean.

3.4 Late Policy

All exams must be taken promptly at the required time. Requests for rescheduling an exam will only be considered if the request is made prior to the start of the exam, or else in an "emergency" situation with appropriate documentation.

Homeworks will always be assigned so that they are due at the beginning of a class meeting. In order to keep students up-to-date on the material (and to allow prompt return of graded material), students are most strongly encouraged to turn in all assignments on time. Printed solution sets will generally be distributed the same day assignments were due.

Late homeworks will be accepted, however with penalty. A late homework will be penalized 20% immediately and an additional 10% for each day that has expired beyond the deadline. Please keep in mind that the policy on Academic Honesty still applies, and thus students submitting late assignments must not use the distributed solution sets either directly or indirectly.

4 Writing Proofs

Much of the submitted work for this course will involve writing formal proofs, and you will be evaluated on the validity and clarity of these proofs.

The "acceptable" level of detail depends very much on what we are asking you to do in a given problem. Anything that is directly related to the "focus" of the particular problem should be proven clearly. In doing so, you may omit proofs of facts which have been seen earlier in this course, or which have clearly been assumed to be part of the prerequisite courses. A reasonable expectation is that your proofs would be clear to a a typical student of this class who is familiar with the course material, but not familiar with the particular problem.

Regarding material covered in prerequisite courses, only assume as "obvious," things which we may safely assume that everyone knows and which are not the focus of the problem you are solving. For example, if you are designing an advanced algorithm and you need to sort n items, then of course you may simply say that this can be done in $O(n \log n)$ worst case time, without explanation. If you have a sorted array of n items and you want to locate an item, you can simply state that you will use binary search, and this required $O(\log n)$ time.

Solutions should be neat and legible. While you should not feel obligated to recopy your work to make it picture perfect, you should do your scratchwork elsewhere.

5 Recommended Readings

In addition to the one required book, the following list contains many other useful sources for information about algorithms and data structures.

- *Design and Analysis of Computer Algorithms*, Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, Addison-Wesley, 1974. ISBN 0-201-00029-6.
- *Data Structures and Algorithms*, Alfred V. Aho and John E. Hopcroft and Jeffrey D. Ullman, Addison-Wesley, 1983. ISBN 0-201-00023-7.
- *Programming Pearls (second edition)*, by Jon Bentley, Addison-Wesley, 1999, ISBN 0-201-65788-0.
- *Computers and Intractability: A Guide to NP-completeness*, by Michael R. Garey and David S. Johnson, W. H. Freeman & Co., 1979. ISBN 0-7167-1044-7.
- *Data Structures and Algorithms in JAVA (second edition)*, Michael T. Goodrich and Roberto Tamassia, Wiley & Sons, 2001. ISBN 0-471-38367-8.
- *Algorithm Design: Foundations, Analysis and Internet Examples*, Michael T. Goodrich and Roberto Tamassia, Wiley & Sons, 2001. ISBN 0-471-38365-1.

- *Approximation Algorithms for NP-hard Problems*, by Dorit S. Hochbaum, PWS Publishers, 1997.
- *The Art of Computer Programming, Volume 1: Fundamental Algorithms (third edition)*, Donald E. Knuth, Addison-Wesley, 1997. ISBN 0-201-89683-4.
- *The Art of Computer Programming, Volume 2: Seminumerical Algorithms (third edition)*, Donald E. Knuth, Addison-Wesley, 1998. ISBN 0-201-89684-2.
- *The Art of Computer Programming, Volume 3: Sorting and Searching (second edition)*, Donald E. Knuth, Addison-Wesley, 1998. ISBN 0-201-89685-0.
- *Randomized Algorithms*, Rajeev Motwani and Prabhakar Raghavan, Cambridge Press, 1995. ISBN 0-521-47465-5.
- *Algorithms in C++, Parts 1–4: Fundamentals, Data Structure, Sorting Searching (third edition)*, Robert Sedgewick, Addison-Wesley, 1998. ISBN 0-201-35088-2. (versions also available for C and Java)
- *Algorithms in C++, Part 5: Graph Algorithms (third edition)*, Robert Sedgewick, Addison-Wesley, 2001. ISBN 0-201-36118-3. (versions also available for C and upcoming for Java)
- *An Introduction to the Analysis of Algorithms*, Robert Sedgewick and Philippe Flajolet, Addison-Wesley, 1995. ISBN 0-201-40009-X.
- *Data Structures and Network Algorithms, CBMS 44*, by Robert Endre Tarjan, SIAM, 1983.ISBN 0-89871-187-8.