Computer Science 344
Algorithms and Complexity
Spring 2008

Dr. Stephen Bloch
office 113A Alumnæ Hall
phone 877-4483
email `sbloch@adelphi.edu`
Web page `http://www.adelphi.edu/sbloch/`
Class Web page `http://www.adelphi.edu/sbloch/class/344/`
office hours Mondays 11–3, Tuesdays 3–4,
Wednesdays 9–12, or by appointment

February 11, 2008

1 Subject Matter

This is a course for anybody who's ever complained about a computer being slow. When faced with a computer program that takes unacceptably long to solve the problem at hand, one option is to buy faster, more expensive computer hardware. But this option has its limits: perhaps your money will run out, or perhaps you'll upgrade to the fastest computer on the market and still be unsatisfied. Often a better option is to find a more efficient approach to the problem — to re-think the *algorithm* so that it makes *better use* of the hardware. A $1,000 personal computer running a good algorithm can often outperform a $1,000,000 supercomputer running a bad (though correct) algorithm.

In this course we'll learn how to measure the efficiency of an algorithm, independent of language, operating system, or hardware. We'll survey a variety of techniques for designing efficient algorithms. (Many of these techniques will help you program correctly even when you're not worried about efficiency.) We'll even learn how to prove that a given algorithm is "as good as it can get," in the sense that *no* algorithm, no matter how clever, will *ever* be better than this one.

Near the end of the semester, we'll study problems believed to have *no* efficient solution by computer program, and even some problems which have *no computational solution at all*, and how we deal with such problems in reality.

2 Textbook

This course will involve reading assignments and homework exercises from the textbook *Analysis of Algorithms: an Active Learning Approach* by Jeffrey J. McConnell. This book is required, and you may find it a useful reference in subsequent courses and software projects. For any of you who are especially interested in this subject, I recommend a more advanced and encyclopedic book, *Introduction to Algorithms* by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. I may also give out other reading assignments by email, on the Web, or in journals.

This is a 3-credit class, which means you should budget 6 hours per week outside class for reading and homework. In particular, you'll need to read about 30 pages per week, on average. *Make time in your weekly schedule for this!*

3 Prerequisites

I assume that everyone in the class has passed CSC-MTH 156 (Discrete Structures) or an equivalent course covering Boolean logic and algebra, graphs and trees, and perhaps recurrence relations. I also assume that everyone in the class has passed at least a year of programming courses; I don't care about the language, as long as you have written and debugged a number of programs and are familiar with the notions of algorithm, recursion, loop, array, linked list, *etc.*

4 Grading

There will be 6 homework assignments, each worth 10% of the semester grade, and a final exam worth 20%. Another 15% of your semester grade will be based on in-class presentations of homework problems; see below. The remaining 5% of your semester grade will be "brownie points", which you by asking and answering good questions in class, coming to me for help when you need it, etc, and you lose by cheating, being a pain in class, etc.

The final exam must be taken at the scheduled time, unless arranged in advance or prevented by a documented medical or family emergency. If you have three or more exams scheduled on the same date, or a religious holiday that conflicts with an exam or assignment due date, please notify me in writing within the first two weeks of the semester in order to receive due consideration. Exams not taken without one of the above excuses will be recorded with a grade of 0.

Homework assignments will be accepted late, with a penalty of 20% per 24 hours or portion thereof after they're due. An hour late is 20% off, 25 hours late is 40% off, *etc.* Any homework assignment turned in more than five days late will get a zero. Any homework assignment turned in after May 6 (the last day of class) will get a zero. (This is so I have, perhaps, time to grade them before the final exam.)

There will be several kinds of homework in this class. At one extreme are the analysis and "thought" problems on paper, resembling problems in a math class. At the other extreme are programming assignments, which may be written in any language that you and I both know and that runs at Adelphi (*e.g.* Scheme, Prolog, Java, C, C++). In between are *pseudocode* assignments: these need to be precise descriptions of an algorithm, like a computer program, but they don't need to meet the syntactic requirements of a compiler (only a human being will read them) and you can ignore details that aren't relevant to the problem at hand. For example, in a problem that wasn't primarily about sorting, you might say "sort table A in increasing order by value" as one line of the algorithm. On the other hand, if the assignment *were* about sorting, I would expect you to give the details of your sorting algorithm.

Each student in the class will present some solutions to homework problems at the board in class, explaining the solution and answering technical questions from me and other students. When I assign a problem, I'll give it a number of "points" based on difficulty (a "point" is roughly one minute's worth of presentation), and you are expected to present 20 "points" worth of problems in the semester. If you find, say, a 3-point (correct) solution to a problem that I rated as a 5-point problem, you get 7 points: 5 for the problem, and 2 extra for outwitting me in elegance and simplicity. If you want to present a particular problem in class, try to get to class a few minutes early and start writing it up on the board. If you're not sure of your solution or your presentation, feel free to discuss it with me in my office before the day you want to present it in class. Your grade for in-class presentations will be my assessment of how well you presented the solution and answered questions about it.

5 Ethics

The Adelphi University Code of Ethics applies to this course; look it up on the Web at `http://academics.adelphi.edu/policies/ethics.php`.

Assignments in this class are to be done individually, or in teams of of two by prior permission; in the latter case, you *may not do multiple homeworks with the same partner.* You may *discuss general approaches* to a problem with classmates, but you *may not copy* large pieces of programs or homework solutions. If you do, *all* the students involved will be penalized (*e.g.* I'll grade the assignment once and divide the points equally among the several people who turned it in).

All work on an exam must be entirely the work of the one person whose name is at the top of the page. If I have evidence that one student copied from another on an exam, *both* students will be penalized; see above.

6 Schedule

This class meets every Tuesday and Thursday from 1:40 to 2:55 PM in Alumnæ 116, except on University holidays or if I cancel class. The schedule of topics, reading assignments, and

homework assignments will be maintained on the Web at
`http://www.adelphi.edu/sbloch/class/344/calendar.shtml`
the dates are subject to change depending on how classroom discussions actually go. I expect you to have read the reading assignments *before* the lecture that deals with that topic; this way I can concentrate my time on answering questions and clarifying subtle or difficult points in the textbook, rather than on reading the textbook to you, which will bore both of us. **Please read ahead!**