University of Michigan Math 416, Fall 2004, Theory of Algorithms

September 6, 2004

1 Infomation

Title:	Math 416, Theory of Algorithms
Time:	MWF, 12-1
Place:	1068 East Hall
Prerequisites: Text:	Math 312 or 412 or EECS 203, and EECS 281 or permission Introduction to Algorithms, 2e Cormen, Leiserson, Rivest, and Stein `http://mitpress.mit.edu/algorithm` (Follow links for errata)
Webpage:	`https://ctools.umich.edu/portal` and follow links
Instructor:	Martin Strauss
Office, central campus:	3063 East Hall, MWF
Office, north campus:	2238 EECS, Tues & Thurs
Email:	`martinjs@umich.edu`
Office hours:	TBA and by appointment

Students with disabilities should obtain a Verified Individualized Services and Accommodations (VISA) form and will be accommodated according to university policy. See
`http://www.umich.edu/~sswd/index.html`

2 Goals

We will learn techniques for analysis of algorithms. This includes language for discussing algorithms separately from their implementation and particular inputs. Is the algorithm correct on all or most inputs? Is it approximately correct? Is the algorithm efficient in terms of run time, memory space, or other resource? Is it efficient on all or most inputs? If the algorithm behaves randomly (often a good thing!), is the algorithm correct or efficient for most of its randomly chosen runs? Will the algorithm be run once, or run many times? In the latter case, do we know in advance the sequence of inputs? After developing the appropriate vocabulary, we study tools for answering these questions with the rigor of mathematical proof. The answer may be negative—there is no algorithm that efficiently solves our problem.

We also design *simple* algorithms and perform *simple* modifications to existing algorithms. Typically the analysis is difficult enough even if the algorithms are simple!

As a secondary goal, to illustrate our techniques, we will study particular algorithms that are fundamental and/or of particular interest to the students. These might include sorting, matrix multiplication, Fast Fourier Transform, and primality testing.

There will be no programming assignments.

3 Required Work

There will be homework assignments, two midterms, and a final. Unless specified otherwise, all work is expected to be completed individually. With proper citation, students may use books and *static* internet pages (excluding, for example, mail to a human expert). When group work is allowed, each student participating in group must have full understanding of all the work submitted by the group.

The final will count for 40% of the grade. Each of the midterms and the combined homeworks will each count 20% of the grade.

You will be responsible for reading electronic announcements which will be posted on a web page and/or sent by email.

Please advise me as soon as possible about any unusual bureaucratic circumstances (*e.g.*, expected absences from exams, special requirements as to grades, etc.)

4 Course Outline

- Basics—non-Randomized
  - CLRS 1-4, Appendices A. (Preliminaries, asymptotic growth, recursive algorithms.)
  - Additional notes (generating functions, sums of binomial symbols)
  - CLRS 6.1–6.4 (Heapsort algorithm, data structures, loop invariants)
- Basics—Probability
  - CLRS 5, Appendix C.1, C.2, C.3, C.5
  - Additional notes (median of means, Cauchy-Schwarz inequality)
  - CLRS 7, 9 (More sorting and selection algorithms, randomized algorithms, average case analysis)
  - CLRS 8.1 (lower bound)
  - CLRS 11.1, 11.2, 11.3, 11.5 (Hashing)
  - Additional notes (random variables of limited independence)
- MIDTERM I (?)
- Meta-algorithmic techniques
  - CLRS 15.2, 15.3 (Matrix-chain multiplication algorithms, dynamic programming)
  - CLRS 16.2, 16.3 (Huffman code construction algorithm, Greedy Algorithms)
- Amortized analysis
  - CLRS 17.1–17.3
- Math problems
  - CLRS 28 (Matrices)
  - CLRS 29.1–29.2 (Linear Programming)
  - CLRS 30.1–30.2 (FFT)
  - CLRS 31 (Number theory: modular arithmetic, primality testing, RSA cryptosystem)
- MIDTERM II (?)
- NP-Completeness
  - CLRS 34.1–34.4
- Approximation Algorithms
  - CLRS 35