Computer Science 445
Analysis of Algorithms
Thomas VanDrunen

Meeting time: MWF 2:00-3:05 pm. (change?)
Meeting place: Amerding 123
Office hours: TTh 1:30-3 pm or stop by whenever my door is open
Final: December 15, 10:30 am.

Textbook: Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms*, Second Edition. McGraw Hill, 2001. *(CLRS)* Campus bookstore: $??; Amazon: $80 new hardcover, $45 new paperback, $13 used.

Course goals: To attain a thorough knowledge of and skill for the methods of designing algorithms and analyzing their efficiency and complexity; to be able to relate analysis theory to practical use in software development and other computer solutions; to have exposure to a wide range of advanced algorithms, data structures, and analysis techniques; to understand the fundamental concepts of NP-completeness.

This course will go fairly slowly at first as we review concepts you may have been exposed to in earlier classes but not thoroughly These need to be mastered for this course to be successful. Then we will pick up pace as we sample a large catalogue of topics; here the goal is more exposure than mastery.

Assessment:

Midterm	25
Final	25
Big programming project 1	15
Big programming project 2	15
14 or so regular assignments	20

The final will be mostly non-cumulative. Specific topics of algorithms and data structures from the first half of the course will not be assessed on the final, but analysis skills of course cannot be forgotten. The programming projects are to teach you how the concepts we talk about theoretically apply (or not!) to real programming, and assess how well you can make those applications. "Regular assignments" will be problem sets (or, occasionally, small programming projects), mostly from the textbook. On average, there will be one per week (which means some weeks there may be none, and some there may be two). Of course I also expect you to attend all class periods and keep up with the readings.

Honesty policy: This course may pose situations where the line between beneficial collaboration and cheating is blurry. Use your best judgment on when helping a classmate or getting help becomes unfair. Discussing the problem sets is encouraged, but you may not share written solutions. Many of the problems are discussed or solved in the computer science literature or on the Internet; using these resources is not appropriate for graded assignments.

Course outline (and approximate schedule):

I. Introduction (week 1; CLRS 1 and 2)
2. Mathematical tools (weeks 2 and 3; CLRS 3 and 4, appendix)
   – Review sets, counting, etc as needed
   – Asymptotics
   – Recurrences
3. Sorting (weeks 4 and 5; CLRS 7, 8, and 27)
   – Basic sorting (bubble, insertion, selection)
   – Advanced sorting (merge, shell, quick)
   – Special-case sorting (bucket, radix)
   – Analysis and comparison

Big programming project (assigned circa 9/20; due c. 9/29)

4. Data structures (weeks 6–8; CLRS 6, 10, 11, 12, 13, 18)
   – Review basic structures
   – Hash tables
   – Trees (BSTs, RB trees, B-trees)

Midterm (10/13 or 10/15)

5. Advanced analysis techniques (weeks 9 and 10; CLRS 15, 16, and 17)
   – Dynamic programming
   – Greedy algorithms
   – Amortized analysis
6. Graph algorithms (weeks 11 and 12; CLRS 22 and 24)
   – Searching
   – Topological sort
   – Shortest path
   – Applications

Second big programming project (assigned c. 11/10; due c. 11/22)

7. NP Completeness (weeks 13 to the end; CLRS 34 and 35)
   – Classification of problems
   – Reductions
   – Well-known NP complete problems
   – Approximation techniques

Final (12/15)