COMP 550 – Algorithms and Analysis

Bulletin Description

Formal specification and verification of programs. Techniques of algorithm analysis. Problem-solving paradigms. Survey of selected algorithms.

General Course Info

Term:	Fall 2017
Department:	COMP
Course Number:	550
Section Number:	1
Time:	Lectures MWF, 10:10 – 11:00
Location:	FB 007
Website:	http://www.cs.unc.edu/~tamert/comp550-f17.html
Piazza:	https://piazza.com/unc/fall2017/comp550001

Instructor Info

Name:	Tanya Amert, Graduate Student Instructor
Office:	FB 132
Email:	tamert@cs.unc.edu
Web:	http://www.cs.unc.edu/~tamert
Office Hours:	TTh 5pm-7pm, or by appointment

Textbooks and Resources

Required Textbook:
For this course, we will use *Introduction to Algorithms, 3rd Edition*, by Cormen, Leiserson, Rivest, and Stein (ISBN: 0262033844). You'll commonly see this book referred to as "CLRS".

All lecture and homework materials will be available on the course website: http://www.cs.unc.edu/~tamert/comp550-f17.html. In addition, programming assignments will be submitted through Sakai.

We are using Piazza as an online forum. You can ask questions about any of the course material on Piazza, but you must refrain from posting solutions to the homework problems, as per the Honor Code policy.

Course Description

In this course, we will first build a framework for talking about algorithms, in terms of both complexity and correctness. Using this foundation, we will introduce a variety of algorithms and the problems they solve, and for each, prove the correctness and time bounds.

Written homework assignments will expand on the theoretical analysis of the algorithms discussed in class, as well as include simple examples. Although the focus of this course is the theoretical foundation of the algorithms discussed, we will have regular programming assignments to reinforce the algorithms themselves, and utilize the data structures you learned about in COMP 410.

Topics: Recurrences, Big-O notation, sorting, searching, graph algorithms, greedy algorithms, dynamic programming, linear programming, clustering algorithms, computational geometry, NP completeness

Target Audience

This course builds on the data structures introduced in COMP 410. We will use the proof techniques from COMP 283 or MATH 381 to analyze and verify the correctness and runtime complexity of the algorithms discussed. In addition, written homework and programming assignments reinforce the concepts through further analysis and implementation.

Given that COMP 410 is a prerequisite, students are expected to have a working knowledge of Java.

Prerequisites

COMP 410, and COMP 283 or MATH 381.

We will use the data structures from COMP 410 as building blocks for our algorithms, and use the proof techniques introduced in COMP 283 and MATH 381 to prove the correctness and runtime complexity of the algorithms we will study.

Goals and Key Learning Objectives

Upon completing this course, students should be comfortable analyzing and implementing a variety of algorithms. In addition, students should be able to determine requirements for a given problem, and be able to choose the proper algorithm to solve the problem.

Course Requirements

The homework assignments will be a combination of written homework questions to give students theoretical practice, as well as programming assignments for practical application. These two parts of each assignment will be labeled separately, with due dates as listed.

During the semester, there will be two midterm exams, as well as a final exam. All exams are cumulative.

Key Dates

First day of class: Wednesday August 23rd
Last day of class: Wednesday December 6th

No class: Monday September 4th, Friday October 20th, Wednesday November 22nd, and Friday November 24th

Midterm exam #1: Wednesday October 4th
Midterm exam #2: Wednesday November 8th
Final exam: Friday, December 15th, at 8:00 a.m.

Attendance and Participation

Although attendance will not be taken in lecture, you are expected to attend all lectures. Participation will also be measured through the use of Piazza, both in asking and helping answer questions. There will be at least 8 in-class pop quizzes, used to gauge the pace and clarity of the material. These will be either anonymous, or used to count towards participation.

Grading Criteria

Homework Assignments	40%
Midterm #1	15%
Midterm #2	15%
Final Exam	25%
Participation	5%

Assignments (written and programming) will be graded both for correctness and style. For written proofs, good style implies that you give enough information to complete the proof, and are not excessively verbose. For programs, this means having well-documented (through comments and/or good variable/function names) and easily readable code. The Google Java Style Guide is a good resource: https://google.github.io/styleguide/javaguide.html.

Late Policy

Each student starts the semester with three late days. These can be used on any homework or lab assignment for any reason, no explanation necessary, and at most two can be used on any given assignment (a single part A or part B). Note that one late day counts as a calendar day, not as 2-3 days until the next class session.

All written assignments will be due at the start of class on the day listed. If turned in after the start of class (including at the end of class), it will count as using a late day. Programming assignments will be due to be submitted to Sakai by the time listed on the assignment.

If you have no more late days, late assignments will not be accepted unless there are extenuating circumstances that you have discussed (in person, or over email and received an acknowledging reply) before the due date.

There will be optional problems on some assignments that count towards gaining additional late days. At most two additional late days can be earned throughout the semester. If you earn these extra late days but do not use them, they will turn into 1% extra credit each at the end of the semester (the original three late days do not count for this).

Course Policies

In class, you are expected to maintain proper etiquette. This includes arriving on time, not having conversations during lecture, and most importantly not having your laptop/phone/newspaper/etc. out during lecture.

Each midterm exam will be 50 minutes in duration, given during a class period. The course final will be 3 hours in duration, and will be given in compliance with UNC final exam regulations and according to the UNC Final Exam calendar.

Honor Code

For written homework assignments, you are allowed to discuss your general approaches with other students, but your final solution must be your own. On the first page of your homework submission, you must list the names of everyone you talked with, and include a signed honor code statement.

For programming assignments, you are allowed to work with up to one other student. You may look at each other's code and help with debugging, but you are expected to type up all of your code yourself. If you worked with someone else, you must each include each other's names in your readme.txt file submitted with your code. See the assignment instructions for more details.

Exams are closed-book, closed-note. You will be allowed an increasing number of "cheat sheets" for the exams: 1 for the first mid-term, 2 for the second, and 3 sheets for the final. Your cheat sheets can be hand-written or typed on 8.5"x11" paper, double-sided, and you are expected to create your own. You are allowed to bring a magnifying glass.

Course Schedule

Week 1	Analyzing and designing algorithms
Week 2	Divide & conquer, recurrences
Week 3	Recurrences cont'd, dynamic programming
Week 4	Dynamic programming cont'd
Week 5	Randomized algorithms, sorting
Week 6	Sorting lower bounds, finding min/median/max, hashing

Week 7	Review + midterm #1, BST review
Week 8	Red-Black Trees, Interval Trees, space partitioning^
Week 9	A* search^, greedy algorithms, knapsack problem
Week 10	Huffman coding, task scheduling, graph review
Week 11	Minimum Spanning Trees, shortest path algorithms
Week 12	Review + midterm #2, Linear programming
Week 13	Linear programming cont'd
Week 14	Convex Hull^
Week 15	Clustering (K-Means, K-Nearest Neighbor)^, NP-Completeness^
Week 16	NP-Completeness^, course recap + final review

^ Some topics may be skipped or replaced depending on time constraints.

Homework assignments will typically be due every other week, with varying proportions given to written and programming questions.

We will not meet for class on Monday September 4th, Friday October 20th, Wednesday November 22nd, or Friday November 24th due to university holidays.

Disclaimer

The instructor reserves to right to make changes to the syllabus, including homework and project due dates and test dates. These changes will be announced as early as possible.