CS 472 - Analysis of Algorithms

CRN 20176 3 Semester Hours

Spring 2022
Waters Hall S105

Instructor:Adam Lewis
Office: Waters Hall S103A
Telephone: 256-233-6505
Email: Adam.Lewis@athens.edu
Google: awadelewis@gmail.com
Class website on Blackboard

Class times:
M 6:00 p.m. - 8:50 p.m.

Office hours:
M: 2:30 p.m. - 5:30 p.m.
T: 1:30 p.m. - 5:30 p.m.
W: 1:30 p.m. - 2:30 p.m.
Other times by appointment.

1 Course Description

This course is an introduction to the classic methods for designing algorithms. The course will study specific problems such as sorting, graph traversals, and matrix multiplication in order to illustrate these methods. The methods studied will include: Divide and Conquer, Back-Tracking, Branch and Bound, and Dynamic Programming and a brief introduction to computational complexity.

2 Prerequisites

Prerequisite: CS372 (Data Structures) and MA308 (Discrete Mathematics)

3 Learning outcomes

Upon successful completion of this course, students will have the ability to:

- Analyze the asymptotic performance of algorithms.
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design techniques and methods of analysis.
- Demonstrate knowledge of computational complexity and understand the limitations of algorithm analysis.

3.1 Texts and Materials

Textbook: **Introduction to The Design and Analysis of Algorithms**, 3rd. ed.
A. Levitin,
ISBN: 978-0-13-231681-1

Additional References: **Lecture notes & Additional reading**
As published on class website on Blackboard.

The Algorithm Design Manual, 2nd. ed.,
S. Skiena,
ISBN:978-1848000698

Forty Algorithms Every Programmer Should Know
I. Ahmad
ISBN:978-1-78980-121-7

4 Class policies

4.1 Department General Course Standards (GCS)

The Department of Mathematical, Computer, and Natural Sciences is committed to a level of course rigor commensurate with those of (1) colleges and universities offering similar mathematics or computer science programs as Athens State University, (2) other Colleges within AthensState University and (3) other departments within the College of Arts and Sciences. The Department of Mathematical,Computer and Natural Sciences supports the content standards, overall quality, and academic integrity for each traditionally delivered, blended or online course within the curriculum according to the general course workload standards delineated below:

1. Reading
2. Writing
3. Course Assessments
4. Class Interaction

4.2 Attendance

If you should be absent for unavoidable reasons, you **must** check with me before I will consider allowing ***any*** make–up work. This includes exams, homework, and project reports.

If you miss a class, you are responsible for getting any material you may have missed.

4.3 Course Grading Scale and Evaluation

A total of 1000 points will be awarded in this course.

Grading Scale:	From 900 pts. or above	A
	From 800 pts. to 899 pts.	B
	From 700 pts. to 799 pts.	C
	From 600 pts. to 699 pts.	D
	Below 600 pts	F

The points will be distributed as follows:

Exam #1	200
Exam #2	200
Exam #3	200
Problem Sets & Programs	400

Grades will posted on Blackboard. Keep track of all your grades. It is your responsibility to make sure they are correctly posted. Any disputes concerning how an assignment has been graded must be brought to my attention no later than the class meeting following the return of the graded assignment.

As some components of your final grade will be computed basis upon a scaled score, the points total on Blackboard may not accurately reflect the total number of points you have collected to date. Please contact the instructor if there are any comments, questions, or concerns about this policy.

4.3.1 Exams

There will be three exams over the course of the semester with two exams during the semester and a final exam. The final exam will *not* be cumulative; i.e., it will cover only the material presented during the final third of the class. The exams will reflect the material presented in class and programming assignments.

Students will be permitted to use one page of handwritten notes (front and back) during exams. Otherwise, all exams are closed book and closed notes.

Exams will be held at the start of the class when scheduled and you will have one hour to complete the exam. Lecture and/or lab sessions will follow the exam. (GCS 1,2,3)

4.3.2 Problem Sets

There will be multiple problem sets. You will be asked to solve from 1 to 10 problems and review questions inspired by the material covered in the previous class. Collaboration is not permitted on problem sets. Each problem set will graded on a scale between 0 and 100 points. (GCS 1,2,3)

Many of the assigned problems will require you to write programs. You may use any programming language to develop solutions. Note that is strongly suggested that you use a language such as C++, Java, Ruby, or Python that has strong support libraries.

At the end of the semester, your score programming assignment score for the semester will be calculated using the following formula:

400*(total points earned / total points assigned)

4.4 Submission of Assignments

Announced assignments **MUST** be completed on the day and time assigned. Late assignments will be accepted with a 25% deduction per day, up to 3 days late.

You will submit assignments using Blackboard. Assignment files (program source, documentation, and others as required) must be submitted as a single PDF file containing design, source code, and documentation. Information about required content and formatting will be provided with the first assignment. Failure to correctly submit files as instructed will result in deduction of points on the assignment.

4.4.1 Academic Misconduct

All acts of dishonesty in any work consitute academic misconduct. This includes, but is not limited to, cheating, plagiarism, fabrication of information, and/or abetting any of the above. Academic

misconduct represents unethical behavior unbecoming to college students. **I have no tolerance for academic dishonestly in any form. Cheating on any assignment, lab, quiz, or exam will result in the student being assigned a grade of F for the course.**

If a student is suspected of cheating, the student will appear before the instructor, as well as other instructors in the computer science department, and explain the reason for the similarities of their code with other student(s) source code.

4.5 Disability Accommodation Statement

If you are in need of classroom or program accommodations, contact the Counseling and Career Services Office/Disability Services, at 256-233-8285 (voice) or 1-800-522-0272, extension 285, or visit the Sandridge Student Center, Room 230 or e-mail **kari.allen@athens.edu** or the Disability Services Office so that your needs can be assessed, documentation can be obtained, and accommodations can be provided/recommended.

4.6 Laboratory Fee

The "lab fee" charged for some Arts and Sciences courses is a fee that provides funding to support and enhance educational opportunities available to students.

4.7 Other

- Check the class website on Blackboard for announcements, postings of lecture notes, quizzes, assignments & solutions, along with labs and the associated solutions.
- Please turn off all cell phones before entering the classroom. I have a very simple policy concerning cell phones: if your phone rings during class, I get to answer it. However, I am fair about this policy; if my phone should ring, I will hand it to one of you to answer for me.

  You may use laptops, tablets or other computing devices in class so long as they do not create a distraction for other students.

4.8 In Conclusion: How to Succeed In This Course

Read this syllabus. Read the textbook, review any extra material on Blackboard, and review the lecture slides before coming to class. Be prepared to ask questions as I prefer class meetings to be a conversation rather than a monologue.

This course moves fast, so I encourage you to keep up with the reading and programming assignments. It can be difficult to catch up if you get behind. If you have any problems or questions, please come talk to me as soon as possible so that I can help.

There may be times when your background has holes relative to your classmates. In these situations, you should certainly seek extra help, but you will at times be expected to do a certain amount of reading and learning on your own.

There may be some of you for whom parts of this course are old material; if this is true, I encourage you to try some of the more advanced, optional parts of the assignments.

Help each other learn. One of the best ways to ensure that you understand a concept is to explain that concept to another person. Note that while you are not allowed to help each other write programs, I encourage you to discuss concepts that are presented in class and in the textbook. For example, a particular programming assignment might ask you to use a particular data structure; you are encouraged to discuss the properties and details of this data structure in the abstract but without looking at other student's code.

Finally, to succeed in this course, it is very important that you learn to think for yourself. For example, you will find that certain aspects of your programming assignments will be under-specified and it will be up to you to think about what the right thing to do is.

4.9 Appendix A: Schedule

Week	Topic
1	**Introduction and Overview** What is an algorithm? **Fundamental Data Structures** Abstract Data Types Set, Bags, and Stuff Trees and Graphs Implementation in C++
2	**Fundamental Data Structures** Vectors, Lists, and Queues Loops and iterators Implementation in C++
3	**Math for Algorithm Analysis** A review of key concepts from discrete math Mathematical Induction and Recursion Logarithms Probability Recurrence Relations
4	***Exam 1*** **Divide and Conquer** Binary Search Mergesort Quicksort Matrix Multiplication
5	**Divide and Conquer** Binary Search Mergesort Quicksort Matrix Multiplication
6	**Dynamic Programming** Graph algorithms Optimization Problems Traveling Salesman Problem
7	**Greedy Alogrithms** Minimum Spanning Trees Dijkstra's Algorithm Scheduling Being Greedy vs. being dynamic
8	**Backtracking** n-queens problem Graph coloring Hamiltonian Circuits
9	**Branch and Bound** Breath-first search A return to the TSP
10	***Exam 2*** **Computational Complexity and Sorting** Computational Complexity

Week	Topic
	Revisiting mergesort and quicksort Heaps and Heapsort Lower-bounds for sorting
11	**Computational Complexity and Sorting** Computational Complexity Revisiting mergesort and quicksort Heaps and Heapsort Lower-bounds for sorting
12	**Searching and Computational Complexity** Interpolation search Searching in trees B-Trees and its variants Adversary Arguments
13	**P vs NP** Input size for problems The sets P and NP Ways of dealing with NP problems
14	**Number-theoretic and Cryptographic Algorithms** A quick (and painless) intro. to number theory. Greatest Common Divisor Algorithms for Modular Arithmetic Finding Large Prime Numbers The RSA Public-Key Cryptosystem
15	**Advanced topics** Parallel algorithms Genetic algorithms
16	***Exam 3***