

CSCI 332: Design and Analysis of Algorithms

[Some details, in red font, still pending]

Course information

Fall 2022

Meeting time: M/W/F 10:00AM-10:50AM

Location: Social Science | Room 344

Course material/submissions/grades are in Moodle (<http://umonline.umd.edu>)

Instructor information

Instructor: Jordan Malof

Office: 4XX

E-mail: jmmalo03@gmail.com

Phone: 406-243-6219

Office Hours:

Mon 230pm-330pm

Wed 430pm-5:30pm

Or by appointment (email me)

Teaching Assistant: TBD

Office: TBD

E-mail: TBD

Office Hours: TBD

Course Objectives

In this course, you will:

- Become fluent in standard techniques for designing algorithms (e.g., divide-and-conquer, dynamic programming, greedy algorithms)
- Learn some of computer science's "greatest hits" algorithms.
- Learn to relate and reduce real-world problems to these fundamental algorithmic problems.
- Learn to reason formally about abstract processes of computation. This includes understanding and applying asymptotics (Big-O), recurrences, induction;

Course Prerequisites

CSCI 232 & (M 225 or M 307). The course builds heavily on the material from these courses, so the prerequisites will be (mostly) enforced.

Required textbook

Algorithm Design

By Jon Kleinberg and Eva Tardos

Circa Summer 2022, the textbook can be found online for free with a simple Google Search.

Other resources

- *Algorithms*, by Dasgupta, Papadimitriou, and Vazirani
(a late draft pdf from 2006 can be found online with a simple search)
- <https://www.khanacademy.org/computing/computer-science/algorithms>
- Algorithm annotations: <http://visualgo.net/>

Schedule

Below is an ordered set of topics we'll cover. It is subject to change. Please consult Moodle for up-to-date schedule and reading assignments. Lectures will cover the reading material as comprehensively as possible. Students are expected to supplement lectures with a careful study of the relevant sections of the textbook.

- Introduction. Chapter 1
- Basics of Algorithm Analysis. Chapter 2
- Review of data structures (arrays, lists, BST, heaps, hashes)
- Graphs. Chapter 3
- Greedy Algorithms. Chapter 4
- Divide and Conquer. Chapter 5
- Dynamic Programming. Chapter 6
- Network Flow. Chapter 7
- NP and Computational Intractability. Chapter 8
- Approximation Algorithms. Chapter 11
- Randomised Algorithms. Chapter 13
- Heuristics

Assignments will include both problem sets (e.g. questions from the text) and programming assignments. A new assignment will be due approximately every 10 days.

Grading

Reading Quizzes	25%
Homework Quizzes	25%
Programming Assignments	25%
Exams	25%

The “curve”: assigned grades are sometimes quite low compared to conventional course grade cutoffs (e.g., ≥ 90 for an A, etc.), and I may adjust these cutoffs at the end of the course. I will provide an update with approximate cutoffs as the semester progresses.

Reading Quizzes

Each 1-2 weeks I will give a short quiz (3-5 questions) at the beginning of class. These will be simple questions about the assigned reading that should generally be easy to answer if you read over the assigned material, in earnest, at least once. The questions will not generally assume that you understood the technical details, but will ask you basic questions about the content. This is a great opportunity to get easy credit in return for effort. Additionally, if you perform the reading on time, you will likely find the lectures more useful, and it will help you avoid getting behind in the course.

Homework, and Homework Quizzes

There will be several assigned homework sets in this course, however, you will not be asked to submit them. Homework solutions can now be found online, usually with significantly less effort than solving the problems. Therefore, I will announce a date on which the homework solutions will be distributed to the class – this is a due date, by which you are expected to have made an effort to solve the problems. Shortly after the solutions are released (e.g., 2-4 days) we will have an in-class quiz that will include 2-3 randomly-chosen homework problems, possibly with minor alterations.

Exams

My exam strategy is as follows: all exam problems will be drawn from homeworks and the questions in the textbook (often with minor modification). Thus the correct way to study for this course is to review these problems and figure out how to solve them. The more you work, the better your grade will be. Exams will be closed book, but I allow one page of notes.

Programming Assignments.

There will be 2-3 programming assignments in this course. These will be submitted and graded. I encourage discussion with others regarding programming assignments, however these should be high-level discussions. Code should be written independently. If I suspect copying or plagiarism, I may ask you to explain each piece of the code to me, possibly resulting in a reduced grade or removal from class.

Late policy

Submissions for programming assignments are due at the beginning of class. Late submissions will not be accepted. Every student will get one free extension on an assignment (programming or homework) for up to a week. You do not have to ask for this – just write that you are using your free extension when you turn it in. Don't waste this extension or feel obligated to use it; another extension will be given only in exceptional circumstances.

Cheating

It should go without saying that academic dishonesty (including plagiarism and cheating) will not be tolerated. Consult the university's student conduct code for more details. I will follow the guidelines given there. I will seek out the maximum allowable penalty for any academic dishonesty that occurs in this course. If you have questions about which behaviors are acceptable, please ask me.

Specifically, do not search for answers to the programming assignments online. I'm not naïve enough to think these don't exist, and students have been caught in the past plagiarizing from the web, often receiving failing grade. I retain the right to question you about the material turned in. If it is evident that you don't understand what you turned in, I may view your submission as an instance of cheating.

Disabilities

Students with disabilities are encouraged to meet with me to discuss *any* accommodations they require.

Electronic devices

Turn off your cellphone, or set it to vibrate during class. Take calls outside the classroom. Students texting during class will be asked to leave. Laptops are permitted, but only for taking notes – if I suspect you are using it for other (esp. distracting) activities, I may ask you to turn it off.

Personal contact

I hope to establish as much personal contact with each of you as is possible. Don't be afraid to attend the "office hours" session, or stop by my office to ask questions or say hello. To facilitate interaction, every few weeks I plan to have a 'Pizza with the Prof'. Outside my office will be a sheet for you to sign-up to join 5-10 other students from the class for a pizza lunch (on me). I look forward to getting to know you.