Syllabus – CSCI232: Data Structures and Algorithms

Spring 2024

Meeting time and location

Lecture: TR 1:30-2:50PM, SS 254

Lab: F 12:00-1:00PM, SS 254
(I will be in the lab until 2:00PM for those who wish to continue working)

Instructor information

Instructor: Doug Brinkerhoff (feel free to use my first name)

Office: Social Science 403

E-mail: doug.brinkerhoff@mso.umt.edu

Office Hours: Office hours will be determined based on availability poll on day 1!

Course Requirements

Prerequisite: CSCI152

Corequisite: M225 or M307

Resources

Website: Course material/submissions/grades are available via Moodle.

Textbook: *Algorithms, Fourth Edition* by Robert Sedgewick and Kevin Wayne

Book Website: http://algs4.cs.princeton.edu
Note that the book website has an abbreviated version of the textbook in html. You should feel free to purchase the text if you like, make do with the abbreviated version, or find a pdf copy online.

Video Lectures: Part 1, Part 2

The expected viewing schedule can be found on Moodle. Note that you'll need to 'register for the Coursera course to access the videos'.

Course Objectives

The purpose of this course is to introduce you to essential data structures and the algorithms that accompany them. These fundamentals will serve as valuable building blocks for the remainder of your career as a computer scientist. We emphasize understanding of both (i) the methods for implementing fundamental data structures and algorithms and (ii) the ways in which these data structures algorithms can be used in code you will write for the remainder of your career. In this course, you will:

- Become familiar with fundamental data structures
- Become familiar with fundamental algorithms based on these data structures
- Improve your software development skills, by implementing these data structures and algorithms in Java
- Become familiar with run-time and space analysis, as applied to algorithm development
- Learn to think like a computer scientist when it comes to finding new efficient solutions to a wide array of problems

Below is a list of topics I expect to cover, in rough chronological order and subject to change. Please consult Moodle for an up-to-date schedule.

- Fundamentals (Objects, data types, APIs, Analysis, Stacks, Queues)
- Sorting (Elementary, Mergesort, Quicksort, Priority Queues)
- Searching (Symbol Tables, Search trees, Hash tables)
- Graphs (Directed and undirected, BFS/DFS, Spanning trees, Shortest paths)
- Strings (Tries, Suffix arrays)
- Compression (Huffman codes)

Course structure

We take a an approach in this class referred to as a "flipped classroom." You will watch video lectures online before class, and come to class with this material already fresh in your mind. Our in-person class time will then be devoted to discussion, individual and group problem solving, and instructor-led clarification of some of the complex ideas in the material. *Before coming to class, you must*

1. watch the recorded lectures and skim the assigned pages in the book. I will assume that you have done so. It will be apparent if you haven't (Why? Because you'll be expected to communicate on these ideas with me and your classmates. Not having a fully crystallized understanding of the material is expected and 100% okay)

2. take the associated reading quiz in moodle.

After class is complete, you should

1. Review any of the lecture videos that remain confusing (perhaps formulate questions for office hours?)

2. Re-read the text in detail

3. Engage with assigned homeworks while this material is still fresh

Most weeks, you will be expected to attend a 1 hour lab section. You should not expect to complete the lab during this time period, and will likely need to finish the activity on your own. However, I will remain in the lab space for an additional one hour to assist those who are available and wish for additional assistance. Specific activities will vary from week to week, but in general you will be expected to implement and experiment with some basic data structure or approach that we have discussed in class. These lab sessions are intended to give you hands-on experience with the structures we care about, and will lay the foundation necessary to succeed in written and programming assignments. You will be expected to submit the results of your in-lab work, usually with a brief writeup.

Grading

The final grade in this course is computed by dividing the number of points earned by the number of points that were possible to earn. Points may be earned by completing tasks in the following categories:

Exams, 50% There will be two non-cumulative midterm exams, held during class time, as well as one final exam. These exams will test your conceptual understanding of fundamental ideas regarding data structures and algorithms. They will not include programming. There will be extensive review sessions prior to each exam. Exams will be graded based on correctness of result (40% of points awarded for getting the correct answer), correctness of approach (30% of points awarded for utilizing a feasible method in the attempt to solve the problem, even if mistakes are made), and clarity (30% of points awarded for clear language, formatting, and handwriting). You may use one page of notes as an exam reference.

Quizzes, 10% Quizzes are designed to ensure that you have completed the relevant reading (and video-watching) prior to the associated class period. They are available in moodle until the beginning of class. Quizzes are multiple choice and will be graded on correctness of result.

Labs: 25% Labs will be a mixture of activities. Roughly half of lab sessions will be used to work individually on shorter one-off assignments that will help you to learn about some particular programming technique, analysis method, or data structure. The other labs will have you work in small groups on a somewhat more challenging programming problem of the type that you might see in the real world. Labs will be graded based on correctness of result (50% of points awarded for getting the correct answer), correctness of approach (10% of points awarded for utilizing a feasible method in the attempt to solve the problem, even if mistakes are made), and clarity (40% of points awarded for clear language, formatting, and code commenting). Note that code that I cannot assign points for code that does not compile: it is not a reasonable use of my time to debug your code.

Homework Assignments: 15% Homework assignments will involve written responses to conceptual questions or step-by-step explanations of the application of critical algorithms. These questions are designed to (1) prepare you for exams and (2) discover the limits of your knowledge and creativity. To the latter end, some of these questions are quite difficult. Homeworks will be graded based on correctness of result (10% of points awarded for getting the correct answer), correctness of approach (50% of points awarded for utilizing a feasible method in the attempt to solve the problem, even if mistakes are made), and clarity (40% of points awarded for clear language, formatting, and code commenting).

Computers

You may develop programs on any machine that you like, but I encourage you to use your own equipment (we can help you in acquiring a laptop from the library if needed). In the first lab, we will help you set up a Java and terminal programming environment under Windows, Mac OS X, or Linux. You may use an IDE if you wish, but it is your responsiblity to ensure that your programs compile from the command line, as this is how I will compile them for grading (historically, I have had substantial issues with automatic import statements in IDEs leading to non-functional code when used on another computer. I do not have the capacity to support debugging of this type).

Late Work

Each assignment will have a due date and time given in Moodle. Assignments must be submitted by that time, and late work will not be accepted. In the event of a technical issue with Moodle, you may E-mail me with your assignment: a lack of Moodle access is not grounds for an extension.

I will allow two late assignments (labs or homeworks) over the course of the semester. You do not need to ask for permission to use these, just note that you are using it when you submit. Assignments with due dates so extended must be turned in prior to the date of the final exam. These are the *only* extensions that I will grant under any circumstances

except for those associated with a request from the Office of Disability Equity or a physician. Documentation is required because I am not qualified to make judgements about whether the severity of your personal circumstances warrants an extension – so please don't ask me to. The best way to ensure that you do not encounter issues with late work is to complete and turn in assignments early (not early in the sense of doing everything in the first week of class, but rather just turn things in a few days in advance of the due date).

Academic Dishonesty

We are in a bit of terra incognita when it comes to academic ethics. The proliferation of widely available language modeling tools like ChatGPT and Bard have made it very easy to produce responses to assessments for the (no offense) pretty easy material that appears in introductory courses such as this one. Simultaneously, we are studying fairly classic problems and so many solutions to, say, homework problems already exist in many forms on the internet, generative models not withstanding.

I am uninterested in trying to root out plagiarism in your code and written responses, and also in tackling the question of whether the usage of chatbots to do your homework is in fact plagiarism. As such, I am explicitly giving you *carte blanche* to utilize whatever resources you may care to in the completion of the labs and homework portions of this course. I hope that you will use these resources in a way that improves your understanding of the material. For example, a reasonable thing to do might be to find an answer to a problem and then study it in detail to ensure that you understand why it works to the extent that you can reproduce such reasoning on a limited-notes exam – which is something that you will have to do.

I strongly encourage you to seek help from me or your TA. You will be pleasantly surprised by how much you (and how quickly) you can legitimately understand a topic with a bit of careful conversation with a real person. Throughout the course, you are also encouraged to work together in small groups. This is because the best way to understand the subtleties of the homework problems is to talk (argue?) about the answers.

Accessibility

The University of Montana assures equal access to instruction through collaboration between students with disabilities, instructors, and the Office for Disability Equity (ODE). If you anticipate or experience barriers based on disability, please contact the ODE at: (406) 243-2243, ode@umontana.edu, or visit www.umt.edu/disability for more information. Retroactive accommodation requests will not be honored, so please, do not delay. As your instructor, I will work with you and the ODE to implement an effective accommodation, and you are welcome to contact me privately if you wish.