Fall 9-1-2021

CSCI 232.00: Intermediate Data Structures and Algorithms

Douglas John Brinkerhoff
University of Montana, Missoula, douglas1.brinkerhoff@umontana.edu

CSCI 232: Data Structures and Algorithms – Fall 2021

Course information

Meeting time/location:
Lecture: M/W 10:00-10:50AM SS 362
Labs: (one of) Th 11:00-12:50 (section 1, SS344)
F 10:00-11:50 (section 2, SS362)

Course material/submissions/grades are in Moodle (http://umonline.umt.edu)

Instructor information

Instructor: Doug Brinkerhoff[1]
Office: Social Science 403
E-mail: douglas1.brinkerhoff@umontana.edu
Phone: 406-243-4597
Office Hours: Mon 11:00–1:00
Wed 2:00–4:00
E-mail for appointment

Teaching Assistant: Tim Anderson
E-mail: tim.anderson@umontana.edu
Office Hours: TBD

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 like stacks, queues, priority queues, associative arrays / hashes, and graphs (e.g. search trees and perhaps tries)
- Become familiar with fundamental algorithms based on these data structures, including sorting, clustering, graph search, and string search
- Improve your software development skills, by implementing these data structures and algorithms in Java

[1] The course material (including this syllabus) is based heavily on previous iterations of CSCI232 developed by Dr. Travis Wheeler.

- Become familiar with run-time and space analysis, as applied to algorithm development

Course Requirements

Prerequisite: CSCI 136/152
Corequisite: M225 (Discrete Math) or M307 (Abstract Math)
Required textbook:
Algorithms Fourth Edition
By Robert Sedgewick and Kevin Wayne
Booksite: http://algs4.cs.princeton.edu/

Flipped classroom

Contemporary research in scientific teaching has shown that lectures are a relatively ineffective means of information transmission, in the sense that if I stand up and talk for an hour, you are unlikely to retain that information in the long term. As such, we take a different 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. While demonstrably more effective, this approach places the following burden on you:

- Before class, 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. Not having engaged with it at all is not)
- there will be quizzes in moodle due before class that you are expected to complete.

After class is complete, you should

- Review any of the lecture videos that remain confusing (perhaps formulate questions for office hours?)
- Read the text in detail

Video Lectures

Recorded lectures are available within these Coursera courses:
Part 1: https://www.coursera.org/learn/algorithms-part1
Part 2: https://www.coursera.org/learn/algorithms-part2
The expected viewing schedule can be found on Moodle. Note that you'll need to "register" for the Coursera course to get access to the videos.

Lab

Most weeks, you will be expected to attend a 2 hour lab section. 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.

Topics

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)

Grading

Assignments will include both problem sets (questions requiring written answers) and programming assignments.

Problem sets: 25%
Programming: 25%
Labs: 15%
Exams: 25%
Quizzes: 10%

Adaptive grade thresholds (The "curve")

You may have heard that the grades assigned in this course for assignments and exams are often quite low. I account for this, and set grade cutoffs accordingly. Cutoffs are usually lower than the typical 90/80/70 splits. I will provide an update with approximate cutoffs as the semester progresses.

Exam Schedule

There will be two mid-term exams contributing equally to the total 25% exam component of the course grade. The exam will be in the form of a one-on-one meeting with me, where I will ask you to expound upon various topics related to the course materials.

Midterm dates (approximate):

- during the week of Oct 04-Oct 08
- during the week of Nov 15-Nov 19

Cheating

Academic dishonesty (including plagiarism) will not be tolerated. Cheating hurts all involved:

- It devalues the grades earned by others in the class, and the degree from our program
- It leaves you without the skills you've asked (and paid) me to help you gain

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 what constitutes acceptable use of resources, I encourage you to reach out to me – I will always respect your attempts to understand the ethics of the situation. With specific regard to answers appearing online, I'm not naïve enough to think these don't exist, and will be on the lookout for plagiarized submissions. Also, don't copy solutions from your classmates. I retain the right to question you about the material turned in. If it is evident that you don't understand it, I will reduce your score, and may treat your submission as an instance of cheating.

Rather than cheat, I encourage you to seek help from me or Tim. You will be pleasantly surprised by how much you (and how quickly) you can legitimately understand a topic with a bit of careful conversation. 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. Read below for a few examples of how I would like you to interact with your classmates.

Working together (problem sets)

I expect that most of you will end up talking about class assignments with other students – that's good! However, each of you should work on all of the problems independently, and not just sub-divide the questions among group members. You are welcome to discuss problems and collectively devise solutions at the conceptual level, but you should not share the way you've written up your solution – each of you should independently write up a separate submission. Do not write your solutions up then share them with someone else. Though the ideas behind your solutions may be similar, the text should be your own.

Working together (programming assignments)

I encourage discussion with others regarding programming assignments, as well. As with problem sets, these should be high-level discussions. Code should be written independently. If I suspect copying or plagiarism, I will ask you to explain each piece of the code to me, possibly resulting in a reduced grade or removal from class.

Working together (labs)

The lab sessions are where you have direct, hands-on exposure to implementing and using simple algorithms and the data structures they depend on. In this capacity, I want you to talk to each other, discussing the most minute details. This can (and should) involve looking at the code of your classmates. The goal is that you learn the material and share what you've learned with others, and I'm not worried about plagiarism here. The exception is that when a lab asks that you write something about what you've done, this should be in your own words!

Late policy

Submissions for programming and homework 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, as another extension will be given only in exceptional circumstances.

Attendance

Attendance is required. You are responsible for all material presented in class, and some of that material is not covered in the textbook.

Computers

You may develop your programs on any machine that you like: we encourage you to use your own equipment. In the first lab, we will provide instructions for setting up a Java and terminal programming environment under Windows, Mac OS X, and Linux. Windows laptops will be available during lab sections.

Disabilities

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

Personal contact

Don’t be afraid to visit my office hours, or stop by my office to ask questions or say hello. In fact, as your first assignment, I require that you do so at least once in the first four weeks of class. Expect to spend 15 minutes.