SJSU SAN JOSÉ STATE UNIVERSITY

College of Science · Computer Science

Data Structures and Algorithms Section 04 CS 146

Spring 2024 3 Unit(s) 01/24/2024 to 05/13/2024 Modified 02/09/2024

Contact Information

Instructor: Dr. David Scot Taylor

Email: david.taylor@sjsu.edu
Office: MacQuarrie Hall 212
Phone: (408) 924-5124
Website: www.cs.sjsu.edu/~taylor

Office Hours

Monday, 12:00 PM to 1:00 PM, MacQuarrie Hall 212

Tuesday, 10:30 AM to 11:30 AM, MacQuarrie Hall 212

by appointment
MacQuarrie Hall 212 or https://sjsu.zoom.us/j/89560963006

Course Information

Lecture, Section 4

Monday, Wednesday, 9:00 AM to 10:15 AM, Interdisciplinary Science 878

Lecture, Section 5

Monday, Wednesday, 10:30 AM to 11:45 AM, Interdisciplinary Science 878

Course Description and Requisites

Implementations of advanced tree structures, priority queues, heaps, directed and undirected graphs. Advanced searching and sorting techniques (radix sort, heapsort, mergesort, and quicksort). Design and analysis of data structures and algorithms. Divide-and-conquer, greedy, and dynamic programming algorithm design techniques.

Prerequisite(s): MATH 30, MATH 42, CS 46B, and [(CS 48 or CS 49J) if CS 46B was not in Java], each with a grade of "C-" or better; Computer Science, Applied and Computational Math, Forensic Science: Digital Evidence, Software Engineering, Data Science majors only; or instructor consent.

Letter Graded

✱ Classroom Protocols

Other than the beginning of the semester (to establish that you are in the class), and exams, I will not take attendance. Nevertheless, you are expected to attend class: The class was designed to be taught in a `flipped' style: most new material will be introduced through videos, as part of homework. During class time, we will try to spend our time on interactive activities, rather than me lecturing to you. Many students feel like this style doesn't teach them as well. Research shows the strong possibility that it is more effective, even if students don't feel that way. Honestly...it feels pretty strange from my side too.

Class partiticipation and feedback are very important to keep the course interesting. *If I am covering material too slowly or quickly, or if I am not clearly explaining things, you must let me know.* I prefer an interactive learning environment. If you disagree with something I say, speak up. Argue with me in front of the class. It will make the class better, and right or wrong, constructive interaction will not not hurt your grade. If you are correct, clearly my mistake should be corrected. If you are incorrect, probably I have not explained something clearly anyway, and at least half of the class is confused by it. Point it out right then and there. In cases of exceptional participation that seem to benefit the class as a whole, I reserve the right to improve a student's grade by up to 1/3 grade.

Program Collaboration Policy

You are expected to code your own programs, with at most minor help from others. Talking abstractly, while still trying to figure it out, with somebody else who is in the same situation, is fine. Sharing code is not, and this includes reading their code and retyping it, or having them dictate it to you. Do not look for premade solutions. **Do not copy code.** You should understand what your code does. If I ask you what something does in your code, and you don't understand why it is in your code or what it does? That is unacceptable, as it indicates that you are submitting work which is not your own. If you can get somebody to explain something to you in detail, to the point that you can understand and code it, that is okay. **Your code will be checked for correctness, but graded on your ability to answer questions about it. It is possible to get credit for code that doesn't work. It is possible to not get credit for working code if you don't seem to understand what it does. This latter case may also be deemed academic dishonesty.**

Although talking with others who are working on the problem is fine, do not directly tell somebody else how to do it if you have already figured out an algorithmic aspect and have yours working. If I think it is appropriate to give algorithmic hints, I will give them, that is my role as the instructor. They should be asking me, not you. **Beyond scheduled office hours, I generally spend over 100 hours this semester answering emails and holding additional office hours. Answering questions, based on the students situation, is a large part of my job as an instructor.** I might not always give as direct an answer as you

are hoping for, it depends on how far you have gotten, what you have tried, and what the deadlines are. But, I really do try to give you answers that will push you towards learning the material. Do not get the answers from someone who has already finished the programs.

I run a code plagiarism check at the end of the semester. Having conversations about copied code with students, and writing reports on academic integrity is one of the few parts of this job that I hate. Please don't put yourself, or me, into this position. There will be an assignment on Canvas to state that you understand the homework and program policy, and that you will not share code, nor use someone else's code.

Recording Lectures or Sharing Course Materials

You can make audio recordings of class for your own personal use. Perhaps you want to want to have my dulcet tones lull you to sleep at night instead of only during class, that is fine. Weird, but fine. Perhaps you want to torture your neighbors by blasting it on your porch, that is not fine: aside from possible violations of the Geneva Convention, recordings should not be reproduced, distributed, or publicly broadcasted. If you want to make video recordings, please discuss it with me.

Course material developed by the instructor is the intellectual property of the instructor and cannot be shared publicly without his/her approval. You may not publicly share or upload instructor generated material for this course such as exam questions, lecture notes, or homework solutions without instructor consent.

Program Information

Diversity Statement - At SJSU, it is important to create a safe learning environment where we can explore, learn, and grow together. We strive to build a diverse, equitable, inclusive culture that values, encourages, and supports students from all backgrounds and experiences.

Course Goals

We will examine various ways to represent data used by programs and to compare these representations in terms of their memory requirements and the resulting program execution times.

Time will be spent learning algorithms and data structures, mathematical tools and techniques (recursion, recurrence relations) useful for their design and analysis, and seeing some examples of when they are needed.

Objectives:

- To ensure that students are familiar with ways to implement elementary data structures and their associated algorithms.
- To introduce students to the implementation of more complex data structures and their associated algorithms.
- To acquaint students with advanced sorting techniques.
- To teach students how to determine the time complexity of algorithms.

- To introduce students to algorithm design techniques.

Course Learning Outcomes (CLOs)

Upon successful completion of this course, students should be able to:

- Understand the implementation of lists, stacks, queues, search trees, heaps, union-find ADT, and graphs and be able to use these data structures in programs they design
- Prove basic properties of trees and graphs
- Perform breadth-first search and depth-first search on directed as well as undirected graphs
- Use advanced sorting techniques (heapsort, mergesort, quicksort)
- Determine the running time of an algorithm in terms of asymptotic notation
- Solve recurrence relations representing the running time of an algorithm designed using a divide-and-conquer strategy
- Understand the basic concept of NP-completeness and realize that they may not be able to efficiently solve all problems they encounter in their careers
- Understand algorithms designed using greedy, divide-and-conquer, and dynamic programming techniques

Course Materials

Introduction to Algorithms

Author: Cormen, Leiserson, Rivest, and Stein
Publisher: MIT Press
Edition: 3rd
ISBN: 0262033844
Availability: widely available

The ISBN-10 is given above, the ISBN-13 is 978-0262033848

This textbook is very widely used, and I hope it will come in handy beyond this course. The 3rd edition, for the material we cover, is quite similar to the 2nd edition. (The 2nd edition managed to obfuscate a few issues from the 1st edition while clarifying others.) I think the majority of changes from the 2nd to the 3rd edition are in sections we don't cover, though some of the exercises and readings have changed. When possible, I will post assignments for both the 2nd and 3rd editions of the book.

You can find errata (bug reports) for the book http://www.cs.dartmouth.edu/~thc/clrs-bugs/bugs-3e.php, for whichever printing of the book you get.

A 4th edition of the book was released in 2022. I am sure that it would be fine as well, I don't think it differs greatly for the material that we cover, though it might be organized differently.

Computer

You need to have access to a computer with Java and a development environment. Additionally, some days you will be expected to have access to some kind of computer/tablet/phone for communication during some classroom exercises, and to a wireless laptop for exams.

Course Requirements and Assignments

The following will be regularly assigned for time outside of class:

- Video lectures
- Rote homework problems or video quizzes given in Canvas.
- Readings from textbook or handouts
- Canvas Quizzes covering other material (this has replaced written exercises of the past, but perhaps sometimes there will still be written questions for submission)
- 4 Programming assignments
- 2 Practice Exams

In addition to these, there will be two days of exams during the semester, a final exam, and several meetings scheduled for in-person code review.

During the introduction of new material, homework is our chance to learn by making mistakes. It is expected that you will make an effort in all of the above for the sake of learning, and to give yourself feedback about your understanding the material.

The purpose of the rote homework is to give you enough practice working on problems to either understand how to solve the problems, or at least to learn from solutions to those problems.

I encourage you to watch each video closely and carefully, pausing and taking notes if needed, before attempting homework on that video. If you don't do well on your first attempt at the homework, it indicates that you have not watched the video closely enough. This should give you some immediate feedback on whether or not you are watching the videos closely enough.

Canvas quizzes on the videos can (usually) be taken up to 5 times, but it is really your first attempt that will give you the most feedback on how well you understand the material. Using what would be a terrible homework assignment to give an example of the difference between the first attempt and later attempts: imagine that you are told to study the first 15 digits of pi. After doing that, if you were asked "What digit is in the is the 10 millionths position?" and you could answer that, it would give some indication that you had memorized the requested digits. On the other hand, if you got it wrong, and then looked at the digits of pi before taking a second attempt to answer the question correctly after memorizing the single digit requested, it would give little confidence that you knew any other digits of pi. Hopefully, your homework will be much more interesting than that example.

If, even **after a second attempt** at a Canvas homework, you still cannot get the correct answers, you can ask a classmate or somebody else to help you. **You may not simply ask them for the answers. (That counts as academic dishonesty.)** You can, on the other hand, have them walk you through the problem,

step by step, until you get to an understanding of the answer. (For this course, that counts as diligence. You should understand your submitted answers, otherwise you need to spend more time on them.) Because the homework has a time limit for submission, you might want to take your 3rd attempt, if needed, to record the questions, so that you can do them with more time on your 4th attempt. You really shouldn't need a 5th attempt. The reason attempts are limited is to stop anybody from making blind submissions until they happen to get the correct answers.

In Spring 2021, I shifted away from written homework problems, but if I do have any, they will not be graded on correctness, but on whether or not it looks like an honest effort attempt was made to answer that problem. **You may discuss problems with others, but anything other than superficial comments must be documented.** You should not simply copy solutions, nor look for solutions (on the web or elsewhere), but if needed you can have somebody explain a problem to you in full, until you understand the solution. I might only return solutions for those problems for which you turn in evidence of putting in enough effort.

Note: The first homework assignments will be available until the end of the week of add/drop date, while the class enrollment is still being worked out. After that, homework will only be available until it is due. If you submit the homework, you will still be able to see it afterwards, but if you don't get around to looking at your homework before its due date, it won't be.

For both Canvas and written homework, you should do each homework, unless you are positive that you understand the topic so well that doing the homework would not be a good use of your time. And for those students? They should be the ones helping classmates to understand the material, as outlined in the two preceding paragraphs.

Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course related activities, including but not limited to internships, labs, and clinical practica. Other course structures will have equivalent workload expectations as described in the syllabus.

✔ Grading Information

Homework:

Homework is for learning material, and for giving you feedback. Grading homework in the traditional sense encourages students to focus on grades more than learning the material, which is not considered to be the best way to learn. Because of this, for this class, an individual's homework scores will not have a direct effect on their grade as an individual. They will at most have a small, indirect effect, as described below. Of course, proper effort on homework by an individual will likely result in higher exam scores for that individual.

Instead, homework scores may be used, slightly, to set the curve for the exams: if the class as a whole is doing its homework, it might push the median test score towards the B- range, while if the class, as a whole, is not doing homework, it might push that median towards the C+ range.

I do **not** believe in assigning homework for the sake of assigning homework. If you understand the material without doing the homework, and the homework seems like a waste of time to you? In that case, your advanced understanding should be reflected by your performance on the rote tests (described below), with the expectation that you should do well on those tests. If you are in the top half of the class on those exams, your homework score will be ignored when setting the curve for the class, so your lack of homework will not hurt anybody. You are encouraged to help your classmates to learn the material, teaching it to them will help you to understand it even more.

Note, **this policy should highly discourage students from cheating on homework:** the (admittedly small) risk of being caught is balanced by the fact that your individual homework score will not help you individually, it will only play a very small role in setting the curve for the exams. Because your homework is only one of many homeworks used to set that curve, in cheating, you are taking all of the individual risk, while not reaping any individual reward. Additionally, you won't actually learn the material.

Due to the way that Canvas homework (multiple attempts) and written homework (graded on effort) are graded, the students making a real effort on homework should, as a whole, expect to get credit for a high percentage (90%?) of homework problems. Why should you do the homework? To learn the material. **Do not interpret homework to be optional. Think of it as mandatory.** It just doesn't happen to be part of your weighted course grade. Musicians need to practice when they aren't performing, athletes need to train when they aren't competing, and you need to study this material even when it isn't graded.

Rote Tests:

There will be two in-class tests in the final weeks of the semester, that will test "rote" knowledge. You will be given a template for each exam a week in advance. These two exams combined will be scored on a (curved) scale from F to B-. (For the purpose of the class, you will just get a letter grade on the exam.) The curve will be based on both test performance, and my impression of the class performance as a whole (from homework and class interaction). **You are expected to have a laptop computer, with the lockdown browser installed, that you can use for the exams.**

Although the grade cut-offs may change as the exams change, the grade cut-offs should be no more stringent than:

65%+ B minus

60-65% C plus

55-60% C

50-55% C minus

45-50% D plus

40-45% D

35-40% D minus

-40% F

Final Exam:

- For Section 4, the final exam will be Thursday, May 16, 7:15-9:30am
- For Section 5, the final exam will be Wednesday, May 15, 9:45-noon

Your final exam will be a mixture of rote and advanced questions. I will give you some practice problems (during the semester) to give you some idea of what that means, but you will **not** get a template for the exam. It will be graded on a curve, from F to A+. **You are expected to have a laptop computer, with the lockdown browser installed, that you can use for the final.**

Although the grade cut-offs may change as the exams change, the grade cut-offs should be no more stringent than:

75%+ A plus

70-75% A

65-70% A minus

60-65% B plus

55-60% B

50-55% B minus

45-50% C plus

40-45% C

35-40% C minus

30-35% D plus

25-30% D

20-25% D minus

-20% F

Programs:

You are expected to work on all programs, to submit them, and to be able to explain them to me. You will have individual code review with me to discuss your code. (For the semester, three regular class meetings will actually be canceled, to make more time for code review, for the first two programs.) Different programs are worth different numbers of points (2, 4, 3, and 2 respectively), for 11 points total. For each program, if you are able to program enough of the work, and explain it to me, you might still get full credit for the program, even if it does not answer every test case. A program that works perfectly that you cannot explain will not get credit, and may violate academic honesty policies.

Course grade

First, I will take the maximum of your Rote Test grade, and you Final Exam grade, to get your exam grade. If you have at least 8/11 points from your programs, that will be your course grade. From 5.5/11 up to but not including 8/11 points will deduct 1/3 of a letter grade. From 3/11 up to but not including 5.5/11 will deduct 2/3 of a letter grade. Finally, under 3/11 points will deduct a full letter grade. The resulting modified exam score will be your course grade.

University Policies

Per University Policy S16-9 (PDF) (http://www.sjsu.edu/senate/docs/S16-9.pdf), relevant university policy concerning all courses, such as student responsibilities, academic integrity, accommodations, dropping and adding, consent for recording of class, etc. and available student services (e.g. learning assistance, counseling, and other resources) are listed on the Syllabus Information (https://www.sjsu.edu/curriculum/courses/syllabus-info.php) web page. Make sure to visit this page to review and be aware of these university policies and resources.

Course Schedule

Date **Subject to change**	Planned Topic	Notes
January 24	Introductions, Administrivia, Warm-Up	
January 29	Finish Warm-Up, Start Application	
January 31	Defining Problems and Loop Invariants	
February 5	Asymptotic Notation Exercise	
February 7	Heap Lab: code review/lab exercise	**No class today:** individual code review by appointment (through canvas) for everyone either Wednesday February 7, Thursday February 8, or Friday, February 9.
February 12	Recurrence Relations, Recursion Exercise	
February 14	Master Theorem or Sorting Questions? Exercise	

February 19	Quicksort/select Questions? Exercise	
February 21	Problem Lower Bounds, Exercises	
February 26	Finish exercise from previous class, lower bounds by reduction	
February 28	23-Tree Code Review	**No class today:** individual code review by appointment for everyone in the class, today or tomorrow
March 4	Linear Sorts, Exercise	
March 6	23-Tree Code Review	**No class today:** individual code review by appointment for everyone in the class, today or tomorrow
March 11	Graph Exercise	
March 13	Graph Exercise	
March 18	Graph Exercises	
March 20	Graph Exercise	
March 25	Graph Exercise	
March 27	Exercise	
April 8	Dynamic Programming Questions	
April 10	Exercise	
April 15	Rod Cutting, other Dynamic Programming Questions	
April 17	Subset Sum Questions Intro to NP	
April 22	NP	
April 24	NP decision vs. Optimization	
April 29	Review Practice MT 1 Advanced Topic for Fun	

May 1	Review Practice MT 2 Advanced Topic for Fun
May 6	Rote exam 1
May 8	Rote exam 2
May 13	Exams returned, Review for Final.

- For Section 4, the final exam will be Thursday, May 16, 7:15-9:30am
- For Section 5, the final exam will be Wednesday, May 15, 9:45-noon