CS 155 (Section 1): Introduction to the Design and Analysis of Algorithms, Spring 2023

David Scot Taylor
Associate Professor
Dept. of Computer Science
San Jose State University
212 MacQuarrie Hall
Phone: (408) 924-5124 (email works better)
Email: david.taylor "at" sjsu.edu

Spring 2023 office hours:

- Monday, 1:30-2:30 (in person).
- Tuesday, 12:15-1:15 (in person).
- Other times available, especially on Tuesdays. Set up an appointment by email.

Office hours will generally be in-person. When they are virtual, use this zoom link for zoom office hours, and you will enter the waiting room (I will see one at a time in the order you enter, occasionally updating the waiting room through chat about the size of the queue.)

Class Meetings:

- Section 1: MH 223, Mon/Wed noon-1:15

Prerequisite Courses

You must show me that your prerequisite courses have been satisfied. **If you do not show me by Monday, February 13 (a few days before drop date), I will probably drop you from the course.** Further, I will not give out any add codes without first seeing prerequisite proof. You should show me grades for CS146.

Course Format

Most of class time will be spent for lectures and answering questions, with reading and homework assignments assigned for completion outside of class.

Course Website

The course website can be found at http://www.cs.sjsu.edu/faculty/~taylor/term/current/CS155/. This site contains a link to this greensheet, a schedule of classes thus far, and other information and announcements. Grades will be posted on SJSU's Canvas website at https://sjsu.instructure.com/, which may also be used for additional communication.

Course Description

Algorithm design techniques: dynamic programming, greedy algorithms, Euclidean and extended Euclidean algorithms, Discrete and Fast Fourier transforms. Analysis of algorithms, intractable problems and NP-completeness. Additional topics selected from: selection algorithms and adversary arguments, approximation algorithms, parallel algorithms, and randomized algorithms.

Course Learning Outcomes

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

- have a full understanding of various algorithmic design techniques: greedy, divide-and-conquer, and dynamic programming
- understand the general notion of complexity classes, P and NP, completeness and hardness, and the relationships between classes by reduction
- know when to use exact, heuristic, and approximation algorithms
- think recursively for algorithm design

Required Texts/Readings

Textbooks

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.

Introduction to Algorithms, 3rd Edition
Cormen, Leiserson, Rivest, and Stein
ISBN-10: 0262033844
ISBN-13: 978-0262033848
MIT Press, 2009

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.

Other Technology Requirements

Students are expected to have wireless laptops, with Java and a Java IDE installed. If this is not possible, please see me.

Other Readings

I will make any additional reading material as needed, either by a link or by hardcopy.

Course Requirements and Assignments

Workload

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

- Readings from textbook or handouts
- Written homework problems
- Programming assignments (maybe)

- Rote homework problems may be given in Canvas.

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 the material, and to give yourself feedback for your own learning.

For written homework problems, they will not be graded on correctness, but on whether or not enough of an attempt was made to answer that problem. You may discuss problems with others (it should 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.

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.

If we have programs, you are expected to code your own, but can get help from others. Talking is good. 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. If you get help from others, it should be documented in comments. **Do not copy code.** You should understand what your code does. (We can go into program expectations more if we actually have programs this term.)

Class Participation

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.

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' Syllabus Information web page at http://www.sjsu.edu/gup/syllabusinfo/.

Drop Date

Note that for this semester, the last day to drop without consequence is ***Thursday, September 15***, and the last day to add is also ***Thursday, September 15***. After these dates it becomes very difficult to drop or add a class, so be sure you are where you want to be before these dates arrive!

Class Format

This class is scheduled to be an in-person class.

COVID-19 and Monkeypox

Students registered for a College of Science (CoS) class with an in-person component should view the CoS COVID-19 and Monkeypox Training slides for updated CoS, SJSU, county, state and federal information and guidelines, and more information can be found on the SJSU Health Advisories website. By working together to follow these safety practices, we can keep our college safer. Failure to follow safety practice(s) outlined in the training, the SJSU Health Advisories website, or instructions from instructors, TAs or CoS Safety Staff may result in dismissal from CoS buildings, facilities or field sites. Updates will be implemented as changes occur (and posted to the same links).

Grading

Each of two exams during the semester will count as 25% of your grade. The final exam will be 50%. Exam curves may be modified due to how much homework the class, as a whole, attempts (with good effort, rather than correctness, for written homework problems). Furthermore, I may set some additional "final only" score, such that you can still pass the class by doing well enough on the final, regardless of your scores on the two exams during the semester.

Final:

- For Section 1, the final exam will be Wednesday, May 17, at 9:45-noon

Recording Lectures or Sharing Course Materials

You can make audio recordings of class for your own personal use, but they should not be reproduced or distributed. If, for some reason, you want video, please come 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 solutionswithout instructor consent.

Tentative Class Schedule

Below is just a rough outline, by week rather than date. (A few special dates, such as those for exams, are added.) We have no meetings in the 0th week of classes, which start on August 19. Week x.y refers to the yth meeting of the class during week x, where y is 1 or 2 (not 0).

Approximate Date **Subject to change**	Planned Topic	Actual Topic (if different) or more detail
January 25	Introductions, Administrivia, Greedy	
January 30	Greedy/Proofs	
February 1	Greedy/Proofs	
February 6	Greedy/Proofs Start Geometric?	
February 8	Geometric	
February 13	Geometric	
February 15	Geometric	
February 20	Geometric	
February 22	GCD, Strassen	
February 27	FFT	

March 1	FFT
March 6	Review
March 8	Exam 1
March 13	Exam Return, Dynamic Programming
March 15	Dynamic Programming
March 20	Dynamic Programming
March 22	Dynamic Programming
April 3	Dynamic Programming
April 5/td>	Dynamic Programming
April 10	NP
April 12	NP
April 17	NP/Approx
April 19	NP/Approx
April 24	Review
April 26	Exam 2
May 1	Exam 2 returned
May 3	Online/Randomized
May 8	Online/Randomized
May 10	Online/Randomized
May 15	Review

- For Section 1, the final exam will be Wednesday, May 17, at 9:45-noon