CS 155 (Section 1): Introduction to the Design and Analysis of Algorithms, Fall 2018

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

My office hours for Fall 2018

- Tuesday, 9:00-11:00
- Other times available, set up an appointment by email.

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 Wednesday, August 29 (two days before drop date), you might be dropped from the course, if other students have been regularly attending in hopes of getting a space.** 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/fall18/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 ***Friday, August 31***, and the last day to add is ***Monday, September 10***. 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!

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).

Final:

- For Section 1, the final exam will be Thursday, December 13, 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.) Although we usually index from 0 in CS, week 1.1 the first (and only) meeting of the first week of class. Week 2.2 refers to the second meeting of the second week.

Approximate Date **Subject to change**	Topics Covered
Week 1.1	Introductions, Administrivia, Greedy
Week 2.1	146 Review, Greedy, Proofs
Week 2.2	146 Review, Greedy, Proofs
Week 3.1	GCD, Strassen
Week 4.1	FFT
Week 4.2	FFT
Week 5.1	Geometric
Week 5.2	Geometric
Week 6.1	Geometric
Week 6.2	Geometric
Week 7.1 October 1	Review

Week 7.2: October 3	Exam 1
Week 8.1	Exam Return, Dynamic Programming
Week 8.2	Dynamic Programming
Week 9.1	Dynamic Programming
Week 9.2	Dynamic Programming
Week 10.1	Dynamic Programming
Week 10.2	Dynamic Programming
Week 11.1	NP
Week 11.2	NP
Week 12.1	NP/Approx
Week 12.2	NP/Approx
Week 13.1 November 14	Review
Week 14.1 November 19	Exam 2
Week 15.1	Online/Randomized
Week 15.2	Online/Randomized
Week 16.1	Online/Randomized
Week 16.2	Online/Randomized
Week 17.1 December 10	Review

- For Section 1, the final exam will be Thursday, December 13, at 9:45-noon

Last Modified: Aug 25, 2018