CS 146 (Sections 5, 6): Data Structures and Algorithms, Fall 2016

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

My office hours for Fall 2016

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

Class Meetings:

- Section 5: MH 225, Mon/Wed 7:30-8:45
- Section 6: MH 225, Mon/Wed 9:00-10:15

Course Format

Much of this course will be taught in "flipped" format: rather than using classtime for lectures, we will be using class to introduce ideas through problem solving, work on difficult problems, and in a few cases, code review. Reading, videos, rote homework problems, and programming will take place at home.

Course Website

The course website can be found at http://www.cs.sjsu.edu/faculty/taylor/term/fall16/CS146/. This site contains a link to this greensheet, a schedule of classes thus far, and a (soon to exist) tentative schedule of future class topics, along with other information and announcements.

Course Goal and Description

Goal: To 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.

Topics covered will include advanced tree structures, directed and undirected graphs, advanced searching and sorting techniques, priority queues and heaps, dictionaries, and mathematical tools and techniques (recursion, recurrence relations) useful for the design and analysis of data structures and algorithms.

Prerequisite Courses

You must show me that your prerequisite courses have been satisfied. **If you do not show me by Wednesday, August 31 (the second week of classes, the last class meeting before drop date), you**

may be dropped from the course. Further, I will not give out any add codes without first seeing prerequisite proof. You should show me grades for CS46B, Math30, and Math42, or their equivalents on a San Jose departmental course equivalence form. You must have a C- or better in each course. You should also have CS49J if you took CS46B elsewhere in a language other than java. Prerequisite courses and what they covered:
CS46A: Iteration and recursion
CS46B: Stacks and queues, lists, dynamic arrays, binary search trees. Iteration over collections. Hashing. Seaching, elementary sorting. Big-O notation. Standard collection classes.
Math30: Ideally, you would have sequences and series (Math 31), but proof of Math 30 will suffice.
Math42: Sets, logic, proofs, induction, combinatorics, probability, and equivalence classes.

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.

Course 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.

Student Learning Outcomes

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

Workload

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

- Video lectures
- Readings from textbook or handouts
- Homework problems (written) will be regularly assigned
- 6 Programming assignments
- 2 Practice Exams

During the introduction of new material, homework is our chance to learn by making mistakes. **None of the above items will count towards your course grade**, but it is expected that you will make an effort to do homework for the sake of learning the material.

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

University Policies: Office of Graduate and Undergraduate Programs maintains university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc.

You may find all syllabus related University Policies and resources information listed on GUPâ€™s Syllabus Information web page at http://www.sjsu.edu/gup/syllabusinfo/

Grading

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. Based on the number of questions you answer correctly, after these exams, you will have a semester grade from F to B-. This grade will be a lower-bound on your course grade.

Final:

- For Section 5, the final exam will be Tuesday, December 20, at 7:15-9:30
- For Section 6, the final exam will be Thursday, December 15, at 7:15-9:30

Your final exam will be a mixture of rote and advanced questions, with the practice exams from the semester used to give you some idea of what that means. You will **not** get a template for the exam. It will be graded on a curve, from F to A+.

Course grade: Your course grade will be the maximum of your grade from the end of semester exams, and your final exam.

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.

Drop Date

Note that for this semester, the last day to drop without consequence is ***Tuesday, September 6***, and the last day to add is ***Tuesday, September 13***. 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!

Tentative Class Schedule

Approximate Date **Subject to change**	Topics Covered
August 24	Introductions, Administrivia, Nim Warm Up.
August 29	Nim, Shortcuts, Max/Min
August 31	Tournaments and Proofs
September 7	Quadratic Sorts and Proofs
September 12	Asymptotic Notation
September 14	Recursive Proofs
September 19	Recurrences
September 21	Recurrences
September 26	Heap Lab
September 28	Sorting Lower Bounds
October 3	BTree Lab
October 5	BTree Lab

October 10	Practice Take Home Exam
October 12	Exam Review and more
October 17	Graphs
October 19	Graphs
October 24	Graphs
October 26	Disjoint Set?
October 31	Dynamic Programming
November 2	Dynamic Programming
November 7	Dynamic Programming, Take Home Practice Exam 2?
November 9	Test review, Dynamic Programming
November 14	Dynamic Programming
November 16	NP
November 21	NP
November 28	Advanced Topics
November 30	Review
December 5	Exam 1
December 7	Exam 2
December 12	Questions?