SJSU SAN JOSÉ STATE UNIVERSITY

College of Science · Computer Science

Introduction to the Design and Analysis of Algorithms Section 01

CS 155

Spring 2024 3 Unit(s) 01/24/2024 to 05/13/2024 Modified 01/23/2024

Contact Information

Instructor: Peter McGlaughlin

Email: peter.mcglaughlin@sjsu.edu
Office: Duncan Hall 282

Office Hours

12-13:30 Tu/Th
Duncan Hall 282

Course Information

Class location: Clark Building 111

Class times: Tu/Th 13:30-14:45

Course Description and Requisites

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.

Prerequisite: CS 146 (with a grade of "C-" or better); Allowed Majors: Computer Science, Data Science, Applied and Computational Mathematics or Software Engineering; or instructor consent.

Letter Graded

Classroom Protocols

Do NOT share any course material publicly (on Canvas, GitHub, etc.) without permission, including but not limited to lecture notes, lecture videos, passwords, homework/exam solutions, and class meeting links.

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

To develop an in-depth understanding of algorithm design techniques and the analysis of algorithms, and to present a substantial introduction to computational complexity and NP-completeness.

Specific Course Objectives:

- To explore details of using dynamic programming to design algorithms in a variety of areas.
- To determine when a greedy algorithm design strategy is appropriate and to effectively use such a strategy.
- To expose students to classical algorithms of higher complexity than they see in CS 146, such as Strassen's Matrix Multiplication, number theoretic algorithms (the Extended Euclidean Algorithm), a max-flow algorithm, and the FFT and some ways to implement it.
- To develop a thorough understanding of the complexity classes P and NP, including exposure to a proof of NP-Hardness from fundamentals.
- To expose students to analysis of algorithms which are at a greater level of difficulty than in CS 146.
- To introduce students to some more complex areas of algorithms, as selected by the instructor.

Course Learning Outcomes (CLOs)

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

1. Use dynamic programming effectively.
2. Design a greedy algorithm when appropriate, including a proof of its correctness.
3. Follow and use fairly complex graph theoretic algorithms such as a max-flow algorithm.
4. Design simple geometric algorithms involving scanning or divide-and-conquer techniques.
5. Simulate the Euclidean and extended Euclidean classical number-theoretic algorithms.
6. Simulate Strassen's Algorithm for Matrix Multiplication when given pseudocode for it.
7. Explain the Discrete Fourier Transform and simulate the Fast Fourier Transform (FFT) algorithm for computing it when given pseudocode for it.
8. Understand the definition of the complexity classes P and NP and be able to recognize some examples of each.

Course Materials

Textbook:

Cormen, Leiserson, Rivest, and Stein, Introduction to Algorithms, third edition. MIT Press, 2009. ISBN-10: 0262033844 ISBN-13: 978-0262033848

Optional References:

Garey, Michael R., and Johnson, David S., Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.

Knuth, Donald E., The Art of Computer Programming Volume 3/Sorting and Searching (second edition), Addison-Wesley, 1998

Library Liaison

Anamika Megwalu, email: *anamika.megwalu@sjsu.edu*, website: https://libguides.sjsu.edu

Course Requirements and Assignments

The following may be assigned:

- Reading Assignments or Handouts
- In Class Discussions, Activities, and Exercises
- Midterm Exams: there will be two exams during the semester.
- Final Exam: The final exam will be comprehensive for the semester.

Technology Intensive, Hybrid, and Online Courses

All students are required to have access to a wireless laptop (running OSX, Windows, or some version of UNIX), upon which you can install required software. Technology used will include Canvas, programming in Java, and an IDE (Integrated Development Environment).

MYSJSU Messaging

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found on *Canvas Learning Management System course login website*. You are responsible for regularly checking with the messaging system through *MySJSU* on *Spartan App Portal* (or other communication system as indicated by the instructor) to learn of any updates.

Workload Expectations

University Policy S16-9 (http://www.sjsu.edu/senate/docs/S16-9.pdf) states that:

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

Course weightings will be as follows:

0% Weekly Exercises - There will be about 3 ungraded problems assigned each week. These problems are a chance to check your understanding of and ability to apply important concepts from lecture. About 50% of the questions on exams will come from these problems or minor variations on them.

25% Homework Assignments - There will be 5 to 6 homework assignments throughout the semester. The lowest score will be dropped. Homeworks will be more challenging than weekly exercises and will teach you new techniques. You will have 2 weeks to complete an assignment, and may work in groups of up to 3 students. One submission per group. More details on assignments can be found on Canvas.

20% Midterm Exams - tentative dates February 27, and April 16 (in class).

35% Final Exam - Friday May 17, 12:15-14:30

Your course grade will be determined by your final weighted average:

A plus = 97% or higher

A = 93% up to 97%

A minus = 90% to 93%

B plus = 87% to 90%

B = 83% to 87%

B minus = 80% to 83%

C plus = 77% to 80%

C = 73% to 77%

C minus = 70% to 73%

D plus = 67% to 70%

D = 63% to 67%

D minus = 60% to 63%

F = 0% to 60%

Boundary cases count as the higher of the two grades.

Final grades may be curved. Any curve will only benefit students. Details can be found on canvas. I also reserve the right to increase your final grade by 1/3 of a letter grade for class participation.

All students have the right, within a reasonable time, to know their academic scores, to review their grade-dependent work, and to be provided with explanations for the determination of their course grades. See University Policy S20-2 for more details.

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

When	Topic	Notes
Week 1	Introduction	
Week 2	Dynamic Programming	
Week 3	Dynamic Programming	
Week 4	Greedy Algorithms	
Week 5	Greedy, Midterm 1 Review	
Week 6	Midterm 1, Start Divide and Conquer	
Week 7	Divide and Conquer	
Week 8	Computational Geometry	
Week 9	Max-flow	
Week 10	Max-flow Applications	
Week 11	Spring Break	
Week 12	Reductions, Midterm 2 Review	
Week 13	Midterm 2, NP	

When	Topic	Notes
Week 14	Finish NP, Start Randomized Algorithms	
Week 15	Randomized Algorithms	
Week 16	Wrap up, Final Review	