SJSU SAN JOSÉ STATE UNIVERSITY

College of Science · Computer Science

Data Structures and Algorithms Section 09

CS 146

Spring 2024 3 Unit(s) 01/24/2024 to 05/13/2024 Modified 01/21/2024

Contact Information

Instructor: Chinmay Nilesh Mahagaonkar

Office Location: Clark Building 238

Telephone: (408) 210-7475 (email preferred)

Email: chinmaynilesh.mahagaonkar@sjsu.edu

Office Hours: Monday and Wednesday 12:00 PM – 1:00 PM on zoom

Class Days/Time: Tuesday and Thursday 4:30PM - 5:45PM

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

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

- Provide a comprehensive understanding of fundamental and advanced data structures, ensuring students can effectively implement and utilize them in various programming contexts.
- Equip students with the skills to develop, analyze, and optimize complex algorithms, particularly focusing on sorting algorithms and their applications.
- Foster a deep understanding of algorithmic time complexity, enabling students to assess and improve the efficiency of their code.
- Introduce and develop proficiency in algorithm design techniques, preparing students for complex problem-solving in computer science and software development.

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

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

Textbooks/Readings

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.

Course Requirements and Assignments

- Discussions and Participation: Weekly discussion sessions on the canvas portal. Topics will be based on the material covered to date. Active engagement in these discussions is expected.

- Quizzes: A total of six quizzes throughout the semester. Quizzes will include both short programming tasks and written questions. The lowest scoring quiz will be dropped, with the remaining five contributing to the final grade.

- Programming Assignments: Programming assignments are to be done individually, unless otherwise specified. They can be discussed but should be implemented individually. More information is given at the time of the first programming assignment. Never use any code you find on the web, unless I provide it. Some assignments have an oral discussion or examination.
- In-semester Exams: There will be two exams during the semester.
- Final Exam: The final exam will be comprehensive for the semester.

Grading Information

Grading Information
Course weightings will be as follows:

Homework Assignments, Quizzes, Discussions	30%
In-semester Exams (2) (Each)	25%
Final Exam	20%

Exams may be curved (up) to raise their grades if needed.

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.
All students have the right, within a reasonable time, to know their academic scores, to review their gradedependent 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

Date	Topic
25 January	Introduction and Review of Data Structures
30 January	Basics of Algorithm Analysis (Best case avergae case worst case , Recurrecence relation)
01 February	Basics of Algorithm Analysis (Master's Theorem)
06 February	Algorithm Analysis contd. (Master's Theorem)
08 February	Binary search, hashing
13 February	Sorting Revision (Sorting lower bounds, Linear time sorting algos: radix, bucket, counting)
15 February	Heaps and Heap sort
20 February	Heap sort(contd.) + priority queue

22 February	Quick Sort, Quick Select
27 February	Merge Sort
29 February	Divide and Conquer I
05 March	Divide and Conquer II
07 March	MIDTERM 1
12 March	Intro to DP - recursion why DP, Backtracking , Top down, Bottom up
14 March	DP -2 Problems and applications
19 March	Intro to greedy algorithms
21 March	Problems - Interview Scheduling, Knapsack
26 March	Intro to Trees + Tree traversals
28 March	Types Of trees, AVL, Red Black trees
02 April	NO CLASS (Spring Break)
04 April	NO CLASS (Spring Break)
09 April	Intro to Graphs
11 April	Graphs : BFS DFS Topological sort
16 April	Graphs : Single Source Shortest Path + Dijkstra's Algorithm
18 April	MIDTERM 2
23 April	Graphs : Bellman Ford + Floyd Warshall algorithm
25 April	Graphs: MST (Prim’s and Kruskal’s algorithm)
30 April	Graphs: Max Flow (Introduction)
02 May	NP
07 May	NP
09 May	Revision, Summary, Doubts clearing session
16 May	FINAL EXAM