SJSU SAN JOSÉ STATE UNIVERSITY

College of Science · Computer Science

Data Structures and Algorithms

CS 146

Spring 2026 Section 06 In Person 3 Unit(s) 01/22/2026 to 05/11/2026 Modified 01/22/2026

Contact Information

Instructor: Benjamin Poon

Email: ben.poon@sjsu.edu

Class Day/Time: Mondays and Wednesdays, 7:30 AM – 8:45 AM

Classroom: MacQuarrie Hall 523

Office Hours: Monday and Wednesday 8:45 AM – 9:15 AM (after class)

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

Classroom Protocols

Students are requested to use Canvas messaging to contact the instructor or attend office hours.

Lectures are not recorded.

Lecture slides are provided via Canvas, usually immediately prior to the start of lecture.

Lectures may not be recorded by students without prior permission of instructor.

Course material developed by the instructor is the intellectual property of the instructor. Students may not (without prior permission of instructor) publicly share or upload instructor-generated material for this course, including but not limited exam questions, projects, lecture notes, lecture slides, hands-on exercises, or solutions.

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 Learning Outcomes (CLOs)

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

- Understand the implementation of lists, stacks, queues, search trees, heaps, and graphs and 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

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

Technology: Laptop with internet access, an IDE (Integrated Development Environment), and the ability to compile and run Java applications.

Information: Syllabus, programming assignment instructions, etc. can be found on Canvas Learning Management System (https://sjsu.instructure.com). You are responsible for regularly checking the messaging system through MySJSU for updates, or other communication system as indicated by the instructor.

Course Requirements and Assignments

SJSU classes are designed such that in order to be successful, it is expected that students will spend a minimum of forty-five hours for each unit of credit (normally three hours per unit per week), including preparing for class, participating in course activities, completing assignments, and so on. More details about student workload can be found in University Policy S12-3: http://www.sjsu.edu/senate/docs/S12-3.pdf.

Programming assignments may be discussed with peers, but must be completed individually unless otherwise specified. Code you submit must be written by you.

Academic Integrity Policy for AI-Assisted Software: Use of AI-assisted software must be cited as a reference source. Failure to cite the use of any AI-assisted software constitutes a violation of SJSU policy F15-7 and the School of Information's academic integrity policy (https://ischool.sjsu.edu/faculty-handbook-academic-integrity).

✔ Grading Information

Course weightings will be as follows:

- 3 Individual Programming Projects (30%; each 10%)
- Midterm (30%)
- Final (40%)

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

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

- A plus = 97% or higher
- A = 93% 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 grade-dependent work, and to be provided with explanations for the determination of their course grades. See University Policy S20-2 (https://www.sjsu.edu/senate/docs/S20-2.pdf) for more details.

Project Late Policy

Project grades are reduced by 10% per day late. For example, if a project is turned in minutes after an 11:59 PM due date, that is considered 1 day late and the score is reduced by 10%. Submitting a project 10 or more days late will thus receive a 0.

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

The schedule is subject to change with advanced notice given.

Date	Day	Lecture #	Topic
Jan 26	Mon	1	Intro: Syllabus, logistics
Jan 28	Wed	2	Review: Lists, stacks, queues
Feb 2	Mon	3	Loop invariants, insertion sort
Feb 4	Wed	4	Asymptotic notation
Feb 9	Mon	5	Divide and conquer, merge sort
Feb 11	Wed	6	Solving recurrences: substitution and recursion tree methods
Feb 16	Mon		No class - Instructor away
Feb 18	Wed		No class - Instructor away
Feb 23	Mon	7	Solving recurrences: master method
Feb 25	Wed	8	Heaps, heapsort, priority queues

Mar 2	Mon	9	Quicksort
Mar 4	Wed	10	Linear time sorts
Mar 9	Mon	11	Hash tables
Mar 11	Wed	12	Review for midterm
Mar 16	Mon		Midterm
Mar 18	Wed	13	Binary search trees
Mar 23	Mon	14	Red-black trees
Mar 25	Wed	15	Red-black trees
Mar 30	Mon		No class - Spring recess
Apr 1	Wed		No class - Spring recess
Apr 6	Mon	16	Graphs, breadth-first search, depth-first search
Apr 8	Wed	17	Topological sort, strongly-connected components
Apr 13	Mon	18	Disjoint sets
Apr 15	Wed	19	Dynamic programming
Apr 20	Mon	20	Dynamic programming
Apr 22	Wed	21	Minimum spanning trees
Apr 27	Mon	22	Shortest paths: single-source: Bellman-Ford and Dijkstra's
Apr 29	Wed	23	Shortest paths: all-pairs: Floyd-Warshall
May 4	Mon	24	NP-completeness
May 6	Wed	25	NP-completeness
May 11	Mon	26	Review for final exam

May 13	Wed		Final Exam (8:30-10:30 AM)