SJSU SAN JOSÉ STATE UNIVERSITY

College of Science · Computer Science

# Data Structures and Algorithms Section 04

## CS 146

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

## Contact Information


Instructor: Doug Case

Email: doug.case@sjsu.edu


Day/Time: Mondays and Wednesdays 6:00 PM – 7:15 PM

Classroom: MacQuarrie Hall 523

Office Hours: Monday and Wednesday 5:25 PM – 5:55 PM (before class in MacQuarrie Hall 523).

## 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

Communication with the instructor

Students are requested to use the provided email to contact the instructor. I prefer email over Canvas messages and will often see emails sooner.

Classroom Protocol

Course material developed by the instructor is the intellectual property of the instructor. Students cannot publicly share or upload instructor generated material for this course such as exam questions, Programming assignment, lecture notes, lecture slides, hands-on exercises or homework solutions without instructor permission.

## 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:

- Implement lists, stacks, queues, search trees, heaps, union-find ADT (Abstract Data Type), 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 (radix sort, 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.
- Comprehend the basic concept of NP-completeness and realize that they may not be able to efficiently solve all problems they encounter in their careers.
- Comprehend 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

Other technology requirement / equipment / material:

You will need a wireless laptop with internet access. 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).

## 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)at http://www.sjsu.edu/senate/docs/S12-3.pdf (http://www.sjsu.edu/senate/docs/S12-3.pdf).

## Grading Information

Course weightings will be as follows:

- 20% Programming Assignments
- 20% First In-semester exam (Midterm #1)
- 20% Second In-semester exam (Midterm #2)
- 40% Final Exam

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.

Programming assignment grades are reduced by 10% per day they are late. For example, a program is considered one day late if it submitted a couple minutes after the due date's 11:59 PM deadline. So, if a program is submitted four- and one-half days late, then the max score it can receive is 50%. There is no point submitting a program more than 9 days late as it already gets a zero then.

## 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

Here's a breakdown of the course, lecture-by-lecture.

Note: This is a tentative schedule and is subject to change but with fair notice.

| Week | Class Dates | Topics |
|---|---|---|
| 1 | Jan 26 | Syllabus, etc. |
| 1 | Jan 28 | Introduction |
| 2 | Feb 2 | Review (Recursion, Lists, Stacks, Queues) |
| 2 | Feb 4 | Loop Invariants, Quicksort, Selection Sort |
| 3 | Feb 9 | Asymptotic Growth, Notations |
| 3 | Feb 11 | Recurrence Relations |
| 4 | Feb 16 | Master Theorem |
| 4 | Feb 18 | Heaps and Heapsort |
| 5 | Feb 23 | Quicksort, Quickselect |
| 5 | Feb 25 | Sorting Lower Bounds |
| 6 | Mar 2 | Linear Time Sorts (Counting, Bucket, Radix) |
| 6 | Mar 4 | Binary Search Trees |
| 7 | Mar 9 | Balanced Search Trees, Red-Black Trees |
| 7 | Mar 11 | Review |
| 8 | Mar 16 | Exam (Midterm #1) |
| 8 | Mar 18 | Graph Introduction (Representation, BFS, DFS) |
| 9 | Mar 23 | DAGs (Directed Acyclic Graphs), Topological Sort, Strongly Connected Compone |
| 9 | Mar 25 | MSTs (Minimum Spanning Trees), Prim's Algorithm |
| 10 | March 30 – Apr 3 | Spring Break – No Classes |
| 11 | Apr 6 | Disjoint Sets |
| 11 | Apr 8 | Shortest Paths |
| 12 | Apr 13 | Shortest Paths |
| 12 | Apr 15 | Dynamic Programming |
| 13 | Apr 20 | Hashing |

| Week | Class Dates | Topics |
|---|---|---|
| 13 | Apr 22 | Floyd-Warshall |
| 14 | Apr 27 | NP |
| 14 | Apr 29 | NP |
| 15 | May 4 | Review |
| 15 | May 6 | Exam (Midterm #2) |
| 16 | May 11 | Review |
| Final Exam | May 18 5:30 PM | Monday May 18, 5:30 PM – 7:30 PM |