CSE 100 Algorithm Design and Analysis

Catalog Description

Introduction to the design and analysis of computer algorithms. Topics include analysis and implementation of algorithms, concepts of algorithm complexity, and various algorithmic design patterns. Course will also cover major algorithms and data structures for searching and sorting, graphs, and some optimization techniques.

Textbooks

T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein. Introduction to Algorithms, MIT Press, 3rd edition, 2009. ISBN 0-262-03384-4

S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani. Algorithms, McGraw-Hill Education, 2006. ISBN 0073523402

J. Kleinberg, E. Tardos. Algorithm Design, Pearson, 2005. ISBN 9780321295354

Course Objectives/Student Learning Outcomes:

The course introduces the basics of computational complexity analysis and various algorithm design paradigms. The goal is to provide students with solid foundations to deal with a wide variety of computational problems, and to provide a thorough knowledge of the most common algorithms and data structures. Topics include analysis and implementation of algorithms, concepts of algorithm complexity, and various algorithmic design patterns. The course will also cover major algorithms and data structures for searching and sorting, graphs, and some optimization techniques. In addition to learning about all these different topics, by the end of the class, you should be able to model a computational problem, understand the trade-offs in the design of a solution, and be able to implement a software solution to the problem. In the overall picture of the CSE program, this is one of the few classes that will teach you foundational skills to develop solutions for computation problems and the skills to implement the solutions in software. By the end of the course, students will be able to:

1. apply knowledge of computing and mathematics to algorithm design;
2. analyze a problem and identify the algorithm appropriate for its solution;
3. design, implement, and evaluate an algorithm to meet desired needs;
4. have the ability to compare and analyze different algorithms and their usage.

Program Learning Outcomes:

- An ability to analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions;
- An ability to design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline;
- An ability to apply computer science theory and software development fundamentals to produce computing-based solution.

Prerequisites:

CSE 030, CSE 031, MATH 024, MATH 032, where CSE 031 and MATH 024 can be taken concurrently with CSE 100.

Topics:

Asymptotic notation. Divide-and-conquer. Recurrent equations and the master theorem. Space and time complexity. Loop invariants. Linear and binary search. Sorting algorithms: insertion sort, selection sort, mergesort, quicksort, heapsort. Sorting lower bounds. Heaps. Binary search trees. Hash tables with chaining and open addressing. Dynamic programming and greedy algorithms. Graphs: definition and relevant problems (path search, flow, minimum spanning trees).

Class/Laboratory Schedule:

Lecture:

MW 3:30 PM - 04:45, Arts and Computational Sciences 120

Lab:

02: Friday 4:30 PM - 07:20 SE1 138 (Shuo Chen)
03: Wednesday 10:30 AM - 1:20 PM SE1 138 (Wesley Ferreira Maia)
04: Monday 10:30 AM - 1:20 PM SE1 138 (Wesley Ferreira Maia)
05: Wednesday 7:30 AM - 10:20 AM SE1 138 (Wesley Ferreira Maia)
06: Friday 10:30 AM - 1:20 PM SE1 100 (Yi Liu)
07: Tuesday 10:30 AM - 1:20 PM SE1 100 (Yi Liu)
08: Tuesday 1:30 PM - 4:20 PM SE1 100 (Yi Liu)

Midterm/Final Exam Schedule:

This schedule is subject to change, but is tentatively set as follows:

- Midterm 1: February 11, in class
- Midterm 2: March 11, in class
- Midterm 3: April 6, in class
- Midterm 4: April 29, in class
- Final Exam: May 12, 6:30 - 09:30 PM

Assessment/Grading Policy

- Lab Assignments: 15%
- Midterms: 56% (14% each)
- Final Exam: 24%
- Participation: 5%

We will not consider granting Incompletes as grades.
Your lowest midterm score will be dropped.

Earning Participation Bonus

Piazza: https://piazza.com/ucmerced/spring2026/cse100

Active and appropriate use of Piazza, lab, and lecture to discuss course topics, ask & answer questions is the best way to earn participation credit. Be sure to read the guidelines below; failure to follow these policies will result in reduced participation scores.

- You are expected to use Piazza ***before emailing questions to the instructor or TA.*** Your order of support should be in the sequence of Piazza, TA, then Instructor, with reasonable time for response in between each request. Adhering to this policy, emails on course material must contain a link to the corresponding Piazza post. Instructional staff will make all efforts to respond to emails within 48 hours during weekdays. Failure to follow this policy will result in a reduction of the participation grade.
- Please make your Piazza posts public. This allows other students to help answer your questions (and makes it easier for us to address the questions which remain unsolved). Of course, as an exception, please do not share solutions to problems that others may still be working on. When in doubt, you can make the post private. We may make posts private or public (anonymously) at our discretion if we think others may benefit from (or be able to help with) your question.
- You must search before posting. Your question may have already been answered by us or other students in the past. Reading other students' posts will let you refine your question, and gives us more time to answer new questions. Posting a redundant question will affect your participation score.

- Avoid open-ended or vague questions such as: "How do algorithms work?" or "How come the solution to a discussion problem is this?". If you walk us through your thoughts and reference specific lines that you find confusing, we can better address the problem you are facing. Being specific helps us uncover any misunderstandings that you may have.
- Please do not expect TAs to debug your code. This is not a good use of their time, and reduces their availability to help the rest of the class with course material. Part of the goal of this course is to turn you into a great tester and debugger. If you are stuck, the TAs can help you find and fix your bugs, but you should be prepared to show the TA evidence that you have written a test that isolates/demonstrates the issue and stepped through the test to find the area of code with the issue.
- Do not post "+1" as an answer or follow-up. Use the "good question" and "good answer" buttons to show support. Excessive +1 follow-ups clutter the post and make the work of the TAs and instructor more difficult.
- Filming, photographing, recording, or taping the class is strictly forbidden. We would like to maintain a classroom environment where all students feel open to ask questions at any time.

Instructors and Office Hours

Hua Huang (hhuang80@ucmerced.edu)
Ross Greer (rossgreer@ucmerced.edu), Mon 5-6 PM, SE2 209

Teaching Assistants

Shuo Chen (shuochen2@ucmerced.edu), Tues 3-4 PM, SE2 311
Wesley Ferreira Maia (wesleyferreiramaia@ucmerced.edu)
Yi Liu (yliu327@ucmerced.edu) Fri 2:30 PM – 3:30 PM, SE2 1st Floor Lobby

Additional Course Policies

CSE100 is a 4-credit course, which includes 2.5 hours of lecture, 3 hours of lab, and various assignments each week. You should plan on spending at least 6~8 hours outside of lecture and lab on reading and assignments.

Student Responsibilities

Please be sensitive to the learning environment. It is assumed that every student is attending lectures to learn; therefore, anything which distracts any student from learning is not appropriate behavior (for example, conversing during lecture, inappropriate chat comments, etc.). In attempting to keep with a business-like, professional atmosphere, any behavior which would be considered inappropriate in a business setting will be addressed during lecture.

Use of Student Work:

Assignments submitted by students may be used as examples for future students for educational or academic purposes. Names will be removed as possible. You may specifically request to not participate.

Labs

Each lab session consists of two parts.

The first part is a discussion session, which is run by the TA. The goal of the discussion is for you to become more proficient in solving algorithms problems, so this part is like a problem-solving review. This is important to get you ready for the midterms and final exams.

The second part consists of lab programming assignments. This semester, we will be implementing various algorithms in C++. Correspondingly, you need to be quite comfortable programming in C++. If you are not already well-versed in C++ programming, you will need to teach yourself, and do so in a hurry. This will put you at a disadvantage relative to your classmates. All lab assignments should be done individually (see Academic Integrity policy below). Copying even part of the code or Think-Pair-Share activity solutions from the sources like StackOverflow, chegg.com, geeks-for-geeks, etc., or using generative AI models, such as language models or content generators (like ChatGPT, Gemini, Claude, etc.), to produce entire or significant portions of your work is strictly prohibited. We will enforce the academic integrity policy very strictly. Please note that for the lab assignments we run your code through a system to detect similarity with other lab assignments submitted by your classmates, a database of previous five (5) years' submissions, and code available in public repositories mentioned above. The algorithm analyses the structure and flow of the code, so simply changing the variable names and introducing minor changes to input/output will not be effective to defeat it. You would have to modify the code so much that you are better off writing the code on your own. We will also use software to detect and identify AI-generated content to verify the originality of students' work. So, caveat emptor. Please check the instructions in CatCourses for more information and/or consult with the TA.

There will be around 13 graded lab assignments this semester. You can submit the code as many times as you want before the deadline. The last version of the code submitted will be evaluated for grading. Labs are released every Monday and are ***due*** at 11:59pm on the 7th day after they are released (indicated by the Due Date on the assignment page). The labs are ***available*** until 11:59pm on the 14th day after they are released (indicated by the Available until date on the assignment page). The period between the due date and available until date is called the ***grace period***. Labs are graded on the day following the deadline, and the day following the end of the grace period.

The labs are graded with an auto-grader based on a standardized output and the grades will be released every week immediately after grading. Each programming assignment is tested against 10 test cases, and your grade is based on the correct output of those test cases (see Late Policy below for additional information). The labs will provide some test cases for you to test your code, but we will use additional test cases to test your code for grading, so students cannot “fake” the output (doing a simple mapping). If your grade is low, you have the option to resubmit your code for re-grading within the grace period. Again, during that extra time, you may resubmit your code as many times as you want, only the last version will be re-graded with the corresponding penalty (see Late Policy below for additional information).

Late Policy

Unless otherwise indicated, lab assignments are due by the specific due date as indicated in the assignment page on CatCourses. If you submit an assignment late (after the Due Date but before the Available Until date), we will take off 10% of the total value.

Assignments are to be submitted electronically using CatCourses. Lab assignments will be graded the day following the due date, so you should have immediate feedback on how well you did. If your lab assignment received a very low score, this means that your code is not working properly. For those students that want to improve the code, they will get 1 week grace period to do it and resubmit. If you resubmit within this period, you will still get penalized 10%, but you have the chance to improve your grade. For example, if in your original (on time) lab submission you got 90-100% score, it makes no sense to resubmit, since the maximum score for a perfect re-submission will be 90% with the 10% penalty. However, if you got 0-20% in your original submission, it is in your best interest to improve your score and resubmit. Even if you resubmit and get only 50% in the new assignment, you will get a final score of 45% (50% - 10% penalty over your 50% score, which is 45%).

Submission File Format:

ONLY a single C++ file. Failure to do this implies no submission, and you will be assigned a zero (0) for the assignment.

Reasonableness:

The “Reasonable Person Principle” applies throughout this course. This principle simply states that a reasonable request made in a reasonable way shall be reasonably handled by reasonable persons. The TAs and I are reasonable people: we expect that everybody else involved in this class will be as well.

Accommodations for Students with Disabilities

The University of California Merced is committed to ensuring equal academic opportunities and inclusion for students with disabilities based on the principles of independent living, accessible universal design and diversity. I am available to discuss appropriate academic accommodations that may be required for students with disabilities. Requests for academic accommodations are to be made during the first three weeks of the semester, except for unusual circumstances. Students are encouraged to register with Disability Services Center to verify their eligibility for appropriate accommodations.

Academic Integrity Policy

The campus Academic Honesty Policy states: “Academic integrity is the foundation of an academic community. Academic integrity applies to research as well as undergraduate and graduate coursework. Academic misconduct includes, but is not limited to cheating, fabrication, plagiarism, altering graded examinations for additional credit, having another person take an examination for you, or facilitating academic dishonesty or as further specified in this policy or other campus regulations. Cheating is the unauthorized use of information in any academic exercise, or another attempt to obtain credit for work or a more positive academic evaluation of work through deception or dishonesty.

Cheating includes, but is not limited to:

- copying from others during an examination;
- sharing answers for a take-home examination without permission;
- using notes without permission during an examination;
- using notes stored on an electronic device without permission during an examination;
- using an electronic device to obtain information during an exam without permission;
- taking an examination for another student;
- asking or allowing another person to take an examination for you;
- tampering with an examination after it has been corrected, then returning it for more credit than deserved;
- submitting substantial portions of the same academic work for credit in more than one course without consulting the second instructor;
- preparing answers or writing notes in a blue book before an examination;
- falsifying laboratory, or other research, data or using another person’s data without proper attribution;
- allowing others to do the research and writing of an assigned paper (for example, using a commercial term paper service or downloading a paper from the internet); and
- working with another person on a project that is specified as an individual project.

Plagiarism refers to the use of another’s ideas or words without proper attribution or credit. This includes, but is not limited to:

- copying from the writings or works of others into one's academic assignment without attribution, or submitting such work as if it were one's own;
- using the views, opinions, or insights of another without acknowledgment; or
- paraphrasing the ideas of another without proper attribution.

Credit must be given: for every direct quotation; when work is paraphrased or summarized, in whole or in part (even if only brief passages), in your own words; and for information which is not common knowledge. The requirement to give credit applies to published sources, information obtained from electronic searches, and unpublished sources. Collusion is when any student knowingly or intentionally helps another student to perform any of the above acts of cheating or plagiarism. Students who collude are subject to discipline for academic dishonesty. No distinction is made between those who cheat or plagiarize and those who willingly facilitate cheating or plagiarism." More information about the policy and the Office of Student Conduct can be found here: http://studentconduct.ucmerced.edu/

Some students may still have some confusion (albeit the policy is quite clear), in particular concerning collaboration. The following rules are in place to make this issue clearer, from the perspective of my class.

Cheating vs. Collaboration

Collaboration is a very good thing. On the other hand, cheating is considered a very serious offense. Please don't do it! Concern about cheating creates an unpleasant environment for everyone. If you cheat, you risk losing your position as a student in the college. The school's policy on cheating is to report any cases to the university judicial office. What follows afterward is not fun. So how do you draw the line between collaboration and cheating? Here is a reasonable set of ground rules. Failure to understand and follow these rules will constitute cheating and will be dealt with as per university guidelines.

The Simpson's Rule

This rule says that you are free to meet with a fellow student(s) and discuss assignments with them. Writing on a board or shared piece of paper is acceptable during the meeting; however, you should not take any written (electronic or otherwise) record away from the meeting. This applies when the assignment is supposed to be an individual effort or whenever two teams discuss common problems they are each encountering (inter-group collaboration). After the meeting, engage in a half-hour of mind-numbing activity (like watching an episode of the Simpsons), before starting to work on the assignment. This will assure that you can reconstruct what you learned from the meeting, by yourself, using your brain.

The Freedom of Information Rule

To assure that all collaboration is on the level, you must always write the name(s) of your collaborators on your assignment in the beginning of your submission file as a comment.

Computer Science Department Policy on Academic Honesty

As stated in the campus-wide Academic Honesty Policy (AHP), “academic integrity is the foundation of an academic community”. Accordingly, the CSE faculty takes this matter very seriously and has embraced a zero tolerance on this matter. The process described in the following establishes the minimum consequences for violations of the AHP in CSE courses, but repercussions may be more severe for egregious violations. The Computer Science Department Policy on Academic Honesty (“CSE Policy” from now onwards), does not substitute the AHP but rather specifies how it will be implemented when students enrolled in classes offered by the Computer Science and Engineering (CSE) department are found in violation of the AHP. In particular, the CSE Policy defines how the CSE faculty implements the “Instructor-Led Process” described in AHP 802.00.A. This policy and the associated processes have been developed in collaboration with the Office of Student Conduct and the School of Engineering and is jointly implemented by the CSE Faculty, the School of Engineering, and the Office of Student Conduct. The CSE Policy has been in effect since the Fall 2019 term.

Preamble

Computer science education relies on a variety of methods to assess students’ preparation and learning. The term “assignment” shall be interpreted as any method or process resulting in a grade or contributing to the final grade for a class. Accordingly, the term “assignment” used in the following includes, but is not limited to: homeworks, quizzes, in-class exams, take-home exams, programming assignments, software projects, and presentations.

Shared Responsibility

Maintaining an environment where academic integrity is valued and enforced requires commitments by both instructors and students. Instructors will specify what type of collaboration is allowed or disallowed for a given assignment, and students should strictly follow the provided guidelines. When in doubt, students should contact the instructor and ask for clarifications. First Infraction: If it is determined that a student has cheated, plagiarized, or otherwise violated the AHP, the student will receive a 0 (or equivalent grade) for the assignment. As per the AHP, violations will be reported to the Dean of the School of Engineering and the Office of Student Conduct for review of possible violations of the Code of Student Conduct.

Additional Infractions

The School of Engineering keeps a record of all infractions reported by its faculty. If upon receiving a notification it is determined that the student has one or more prior violations of the AHP, the School will inform the instructor who reported the new violation. The additional violation will immediately lead to a failing grade (F) for the course. The student will be informed in writing and will not be allowed to withdraw from the class. According to CSE Policy, students should note that even the first infraction in a class may lead to a failing grade if after reporting it is determined that the student had been previously sanctioned for one or more infractions in

other classes. Students will have the right to appeal the instructor's decision as per AHP 802.00.A.

Resources

If in doubt, students are encouraged to seek guidance from the faculty, advisors, and the Office of Student Conduct. Additional resources can be found on:
http://studentconduct.ucmerced.edu/
https://ombuds.ucmerced.edu/
https://eecs.ucmerced.edu/sites/eecs.ucmerced.edu/files/page/documents/computer_science_department_policy_on_academic_honesty_fall_2019.pdf

Course/Reading Schedule (Tentative):

Each lecture topic has the corresponding chapter coverage indicated.

Week	Monday	Wednesday
1 (Jan 19)		**(Prof. Greer)** Intro (Ch. 1)
2 (Jan 26)	Sorting (Ch 2)	Sorting ctd. (Ch 2)
3 (Feb 2)	Recurrences, Asymptotics, and Master Theorem (Ch 3, 4.4 – 4.6, Dasgupta-Papadimitriou-Vazirani Sec 2.2)	Recurrences, Asymptotics and Master Theorem cont. (Ch 3, 4.4 – 4.6, Dasgupta-Papadimitriou-Vazirani Sec 2.2), Selection and Median (Ch 4.3, 9)
4 (Feb 9)	Selection and Median (Ch 4.3, 9) ***Note: online lecture.***	Midterm 1
5 (Feb 16)	Holiday	Selection and Median cont. (Ch 4.3, 9)
6 (Feb 23)	Maximum-Subarray and Strassen's Algorithm (Ch 4.1 – 4.2)	Maximum-Subarray and Strassen's Algorithm cont. (Ch 4.1 – 4.2), Heaps (Ch 6)
7 (Mar 2)	Randomized Algorithms and Quicksort (Ch 5, 7)	Randomized Algorithms and Quicksort cont. (Ch 5, 7), Bucket Sort, RadixSort, and Sorting Lower Bounds (Ch 8)
8 (Mar 9)	Bucket Sort, RadixSort, and Sorting Lower Bounds cont. (Ch 8)	Midterm 2

9 (Mar 16)	**(Prof. Huang)** Binary Search Trees and Red-Black Trees (Ch 12.1 – 12.3, 13)	Binary Search Trees and Red-Black Trees (Ch 12.1 – 12.3, 13), Hashing (Ch 11)
10 (Mar 23)	Spring Break	
11 (Mar 30)	Graphs, DFS, BFS and Topological sorting (22.1 –22.4, Klienberg-Tardos Ch 3)	Graphs, DFS, BFS and Topological sorting cont. (22.1 – 22.4, KlienbergTardos Ch 3), Strongly Connected Components (Ch 22.5)
12 (Apr 6)	Midterm 3	Weighted Graphs: Dijkstra's and Bellman-Ford Algorithm (Ch 24.1, 24.3)
13 (Apr 13)	Weighted Graphs: Dijkstra's and Bellman-Ford Algorithm cont. (Ch 24.1, 24.3), Dynamic Programming and shortest paths: Bellman-Ford, FloydWarshall (Ch 25.2, 15.1)	Dynamic Programming and shortest paths: Bellman-Ford, Floyd-Warshall cont. (Ch 25.2, 15.1), More Dynamic Programming (Ch 15.4)
14 (Apr 20)	More Dynamic Programming cont. (Ch 15.4)	More Dynamic Programming cont. (Ch 15.4), Greedy Algorithms (Ch 16.1 – 16.3)
15 (Apr 27)	Greedy Algorithms cont. (Ch 16.1 – 16.3), Minimum Spanning Trees (Ch 23)	Midterm 4
16 (May 4)	Minimum Cut and Karger's Algorithm (Kleinberg-Tardos Sec 13.2)	Maximum Flow and Ford-Fulkerson (Ch 26.1 – 26.3)
17 (May 11)	Final Exams	