COMP/MATH 354: Analysis of Algorithms
Fall 2025

THIS IS A DRAFT SYLLABUS WHICH MAY BE UPDATED THROUGHOUT THE COURSE
Last updated: January 8, 2026

Instructor

Michael Soltys
michael.soltys@csuci.edu
Shasta Hall 2611
Office hours: Thursdays 11:30–2:30 or by appointment

Course Information

COMP/MATH 354
Lecture time & place: Wednesdays 6:00–7:00
Prerequisites: MATH 300 (Discrete Math) and some computer programming experience
Co-requisites: None
General Education Areas: None

Catalogue Description

Computer-oriented study of semi-numerical and non-numerical algorithms. Topics include: sorting, tree searching, generation of combinatorial structures, algorithm proof techniques, best algorithms, programming complexity, and string matching. Some computer programming experience required.

Course Details

This course is an introduction to the analysis of algorithms. The material covered consists of five parts: (i) basics of proving correctness of algorithms using pre/post-conditions, loop invariants and termination. (ii) Three classical algorithm design techniques: greedy, divide-and-conquer and dynamic programming (most of the course is dedicated to this part). (iii) Analysis of performance using worst-case complexity. (iv) Implementation issues (we will implement algorithms in Python 3). (v) Finally, we will also mention briefly NP-hardness, and other classes of algorithms such as online, randomized and parallel.

Student Learning Outcomes (SLOs)

Upon successful completion of the course you will be able to:

1. Design algorithms to solve problems according to standard design principles (greedy, divide-and-conquer and dynamic programming);

2. Measure the performance of an algorithm in terms of worst-case complexity and Big-Oh notation, and indicate trade-offs (speed versus memory usage, etc.);

3. Prove the correctness of a given algorithm, i.e., that it solves correctly a given problem.

Course Outline

Topics in order:

1. Correctness: pre/post-conditions, examples of division and Euclid's algorithm. [1 week]

2. Ranking algorithms: PageRank, Stable Marriage, Pairwise Comparisons. [1 week]

3. Greedy algorithms: Minimum-cost spanning trees, especially the idea of a promising solution; job scheduling with profits; other examples. [3 weeks]

4. Divide and conquer algorithms: Mergesort, multiplication of binary numbers, Savitch's algorithm for reachability in little space (Savitch's algorithm is an example of performance where little space requires a long time); other examples. [3 weeks]

5. Dynamic Programming Algorithms: Longest monotone subsequence problem, All pairs shortest path problem, variants of the Knapsack problem and introduction to NP-completeness and approximation algorithms. Illustration of the "overwriting" technique in implementation. Activity selection with profits. [3 weeks]

Textbook

3rd edition of *An Introduction to the Analysis of Algorithms*, by Michael Soltys, published by World Scientific (ISBN: 978-981-3235-90-8). Algorithms and many solutions to problems can be found in https://github.com/michaelsoltys/IAA-Code.

Grading

1. Quizzes: 5% each, 8 quizzes
   Very useful for your own assessment of your understanding of the material, and as preparation for the midterm and to the final exam. To be given weekly.

2. Assignments: 5% each
   Four assignments to be completed individually, and are meant to reinforce the material in a deeper, hands-on, manner; please keep in mind the following:

   (a) First, it is part of solution development to have a back and forth between the instructor and the students, in order to understand fully the requirements and specifications. Thus, you should ask in class if anything about the assignment is not clear, as usually there are many implicit assumptions that must be made explicit.

(b) Second, each solution will consist of a Python program, well documented with comments, submitted using GitHub classrooms. The details on how to do it will be given in class.

(c) Third, it is ok to discuss the assignments with others, but no written notes should be taken out of such discussions.

3. Tests: 30%

   (a) Midterms: 10% each, two midterms

   (b) Final: 20% (cumulative, i.e., containing material from the entire course)

Grade determination

From	To	Letter Grade	From	To	Letter Grade
97	100	A +	77	79.99	C+
94	96.99	A	74	76.99	C
90	93.99	A-	70	73.99	C-
87	89.99	B+	67	69.99	D+
84	86.99	B	64	66.99	D
80	83.99	B-	60	63.99	D−
			0	59.99	F

Policies

1. **Academic Dishonesty:** By enrolling at CSU Channel Islands, students are responsible for upholding the University's policies and the Student Conduct Code. Academic integrity and scholarship are values of the institution that ensure respect for the academic reputation of the University, students, faculty, and staff. Cheating, plagiarism, unauthorized collaboration with another student, knowingly furnishing false information to the University, buying, selling or stealing any material for an examination, or substituting for another person may be considered violations of the Student Conduct Code (located at http://www.csuci.edu/campuslife/student-conduct/academic-dishonesty.htm). Application of the Policy on Academic Integrity (SP.19.001 or current version) to course grades: Instructors shall adhere to the general guidelines in the policy. If a student is found responsible for committing an act of academic dishonesty in this course, the student may receive academic penalties including a failing grade on an assignment or in the course, and a disciplinary referral will be made and submitted to the Dean of Students office.

   The assignments will be written in groups. Each group has to work independently of the other groups; verbal discussions of problems among groups are allowed, but you should not show written notes, and you should not leave such discussions with written notes. Please speak to the instructor if these expectations are not clear.

2. **Disability Statement:** If you are a student with a disability requesting reasonable accommodations in this course, please contact the student disability accommodations office. All requests for reasonable accommodations require registration with the accommodations office in advance of needed services. Faculty, students, and the accommodations office will work together regarding academic accommodations. Students are encouraged to discuss approved accommodations with your faculty.

3. **Basic Needs:** Please use the link to the Basic Needs Program on the Syllabus Policies and Assistance website for information on emergency food, housing accommodations, toiletries, and connections to critical resources.

4. **Syllabus Policies and Assistance Website:** CSUCI's Syllabus Policies and Assistance Website provides important details about academic policies, campus expectations, and student support services that are all highly applicable to your success as a student both in and outside of the classroom. Ensure that you review this site on a regular basis to stay informed about the policies and resources that support your success, as campus resources or policies may change semester to semester.

5. **Final Exam:** This course has a final exam as indicated in the grading section. The final exam will be held in accordance with the approved Final Exam schedule as indicated in the university exam schedule.

6. **Attendance Policy:** Attendance policy as specified in the Policy on Class Attendance (SP.01.056 or current version).

7. **Course Policies Subject to Change:** It is the student's responsibility to check the course's web page frequently to stay abreast of the course, and for corrections or updates to the syllabus. Any changes will be posted there.

Course Assessment

Computer Science Student Learning Outcome (SLO) "1." states:

> *Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.*

Here is the rubric for this outcome:

Performance Indicator	**Unsatisfactory**	**Developing**	**Satisfactory**	**Exemplary**
1. **Algorithmic design:** ***principle of computing***	no understanding of problem, no solution	problem understood, but solution wrong	problem understood and a solution given	problem understood and best solution given
2. **Performance analysis:** ***computational complexity***	no understanding of what is requested	understanding of worst-case but no Big-O estimate	worst-case analysis and a Big-O estimate given	worst-case analysis resulting in tight Big-O estimate
3. **Proof of correctness:** ***Mathematics as other discipline that helps identify solution***	no understanding of how to approach the proof	providing general direction but no details	an outline of the proof given and aspects of framework	a complete proof, with framework of pre/post-condition and invariants

The threshold will be 80%, that is, at least 80% of students must meet the "satisfactory" or "exemplary" level. All three rows will be measured by the corresponding question on the final exam:

A Design Question: A problem is posed, and the students must choose one of the three basic algorithm design techniques to solve it, and present the solution in clear and correct pseudo-code.

A Performance Question: An algorithm is posed, and the student must evaluate its time and/or space complexity in terms of worst-case performance expressed in Big-O notation, and trade-offs, e.g., optimization versus speed, or time versus space resources.

Proof of correctness Question: The student will be given a problem, and an algorithmic solution will be requested, together with the proof of correctness of the algorithm; the student will be required to tie the algorithmic solution to the problem, and to show that the algorithm solves that problem.

This syllabus is written in accordance with CSUCI Senate Policy 24-07 – Syllabus Policy