Course Syllabus

For additional course information, including prerequisites, corequisites, and course fees, please refer to the Catalog: https://catalog.uvu.edu/

Semester: Spring
Year: 2025
Course Prefix: CS
Course and Section #: 3310-001
Course Title: Analysis of Algorithms
Credits: 3

Course Description

Develops and reinforces ability to write and mathematically analyze foundational computer algorithms. Includes formalizing NP-completeness, divide and conquer strategies, greedy algorithms, dynamic programming, backtracking, branch and bound, approximation algorithms and multicore parallelization.

Course Attributes

This course has the following attributes:

☐ General Education Requirements
☐ Global/Intercultural Graduation Requirements
☐ Writing Enriched Graduation Requirements
☒ Discipline Core Requirements in Program
☒ Elective Core Requirements in Program
☐ Open Elective

Other: *Click here to enter text.*

Instructor Information

Instructor Name: Gregory Mortensen

Student Learning Outcomes

1	Calculate the Big-O complexity of an algorithm.
2	Classify problems that embed strategies like divide and conquer, branch and bound, and parallelism according to Big-O complexity.
3	Summarize intractability and its impact on computation.
4	Write sequential and multicore parallel versions of algorithms.
5	Benchmark sequential and multicore parallel versions of algorithms.

Course Materials and Texts

Required materials, fees, and technology

CS 3310: Analysis of Algorithms
zyBook ISBN: 979-8-203-96362-8
Discrete Mathematics, Irani, et al.
Algorithm Design and Applications, Goodrich, et al.

Python 3.13 or compatible version is required.

Technology Expectations:

- Access to the Internet, Teams, and Canvas
- Computer with Python 3.13 installed and the ability to install Python modules (free download)
- Use a code editor of your choice, but all code must run from a terminal window command line interface
- Computer with reasonable graphics display capability

Course Requirements

Course Assignments, Assessments, and Grading Policy

Exams (25% of grade):

- Exams are closed neighbor, friend, forum (you can't post a question on StackExchange), but are open book, open notes.
- Because of flagrant cheating, the midterm and final will both be oral examinations. You must solve your problems F2F, in class, showing your work in front of the class.
    - Your overall exam's score will be as follows:
        - Student completes task correctly and without any prompts:
            - 100% on that question
        - Minor error or uses one prompt:
            - 90% on that question
        - Minor errors and/or 1-2 prompts:
            - 75% on that question
        - Major errors or multiple prompts (3 or more):
            - 0-50% on that question
    - There will be at least two midterms

Projects (40% of grade):

- Linear Search: implement linear search, keep a basic operation counter, using closed form solutions, calculate the average number of executions (basic ops) if the number searched for is in the subset vs the average number of executions if the number searched for might or might not be in the list.
- Binary Search: implement a recursive binary search, keep a basic operation counter, using a closed form solution for the average number of executions (basic ops) for searching a perfect binary tree (one where all nodes have two children and all leaves are at the same level) if x is somewhere in the binary tree.

- Merge Sort: implement a divide and conquer merge sort and keep a basic operation counter. Find a closed-form solution for the best and worst case of merge sort. Use induction to prove the closed-form solutions are mathematically correct.
- Quick Sort: implement a divide and conquer quick sort and keep a basic operation counter. Find a closed form solution to quick sort's best and worst case of quick sort. Use induction to prove the closed-form solutions are mathematically correct.
- Merge Sort vs Quick Sort analysis. Using your code from zyBooks for Merge Sort and Quick Sort, generate a list of random values and compare the performance of Merge Sort and Quick Sort.
- Genetic Sequencing I: code a dynamic programming implementation of gene sequencing and keep a basic operation counter. Create a closed form solution for the time and space complexity of the DP Genetic Sequencing algorithm.
- Genetic Sequencing II: code a divide and conquer implementation of a gene sequencing algorithm and keep a basic operation counter
- Genetic Sequencing III: add memorization to your Genetic Sequencing D&C algorithm
- Genetic Sequencing I, II, and III analysis. Using your code from zyBooks for the various Genetic Sequencing algorithm implementations, compare the performance of the three implementations for both time and space complexity
- Implement the RSA algorithm using the extended Euclidian GCD algorithm, the Sieve of Eratosthenes, and fast modulo exponentiation to both encrypt and decrypt
- Analysis of RSA, the big-O of the GCD function and the big-O of the modulo exponentiation function.

Homework (20% of the grade): there are roughly 16 weekly reading assignments (approx. 1/wk)

Final Presentation (15% of the grade): Find an interesting algorithm, or an interesting application of an algorithm, do a 5 minute in-class presentation about it, using at least 3 references. Discuss: the origin of the algorithm, the authors of the algorithm, why it is important to the field of computer science, big-theta analysis of the algorithm, and sample code snippets.

Required or Recommended Reading Assignments

Week 1: sections 2.1-2.7
Week 2: sections 7.1-7.3, 8.4-8.5
Week 3: sections 8.2, 8.7-8.8, 8.10-8.12
Week 4: sections 8.13-8.17
Week 5: sections 13.1, 27.3, 28.3-28.5, 29.1-29.2
Week 6: sections 9.1-9.2, 9.5, 11.1-11.2
Week 8: sections 27.6-27.7, 28.2, 29.6
Week 9: no reading, continued coverage of week 8's reading
Week 10: Spring Break
Week 11: sections 14.6, 25.1-25.2, 30.1.30.4
Week 12: sections 9.7-9.9
Week 13: sections 41.1-41.5
Week 14: sections 33.1-33.5
Week 15: sections 32.1-32.7

General Description of the Subject Matter of Each Lecture or Discussion

Week 1: Empiric (Statistical) Analysis, Linear Search Project
Week 2: Big-O, Big-Omega, Big-Theta, Binary Search Project
Week 3: Divide and Conquer; Recurrence Relations, Merge Sort
Week 4: Power Functions (Polynomial Time); Induction, Analysis, and Reporting
Week 5: Induction and Recurrence (Cont.), Divide and Conquer Analysis
Week 6: Graph Analysis, Midterm 1 (Oral Exam)
Week 7: Dynamic Programming (DP), Midterm 1 (Oral Exam, cont. if necessary), DP Gen Seq Proj
Week 8-9: DP & D&C Gen Seq (with and without memoization)
Week 10: Spring Break
Week 11: Greedy Algorithms, DP & D&C Analysis Project
Week 12: Backtracking, Tree Traversal, and Branch and Bound (B&B), RSA Project
Week 13: B&B continued; RSA Analysis
Week 14: Polynomial Time (P) vs Non-deterministic Polynomial Time (NP)
Week 15: Combinatorial Algorithms, Traveling Salesperson (TSP)
Week 16: Final Presentation

Required Course Syllabus Statements

Generative AI
ChatGPT (and similar Tools) in This Course: Use ChatGPT as a learning assistant, not as a crutch. If you use it, cite it at the top of your code, note in presentations and acknowledge in written papers. You are responsible to make sure that any code or content does what it is supposed to do and says what you want it to say. Don't accept anything it generates at face value without checking it critically. These days potential employers will expect you to know how to use tools like ChatGPT to generate code and content, so it is a skill we need to teach you. If it helps you learn some things faster, GREAT. Just remember: If you REALLY want to be good, work for it. Does your instructor REALLY expect you to use GEN AI in this class? REALLY? Yes! Suggestions for using it responsibly:

1. Concept Clarification: If you're stuck on a concept like memorization or linear programming GenAI can explain it in simpler terms or provide examples to help you understand better.
2. Practice Problems: GenAI can generate practice problems for you to solve on your own. After you attempt them, it can help you check your answers and understand any mistakes.
3. Study Tips: GenAI can offer strategies for studying various algorithms effectively, such as how to break down complex problems or how to organize your study sessions.
4. Resource Recommendations: GenAI can suggest other textbooks, online courses, or websites that provide additional explanations and practice problems.
5. Homework Guidance: While GenAI could but should not do your homework for you, it can help you understand the questions and guide you on how to approach solving them.
6. Exams: GenAI is ***not*** to be used on Exams. For remote exams such as in Canvas, your instructor leaves people on their honor not to cheat in any form. Since the midterms are in class, GenAI isn't a problem for this class.
7. Debugging and Understanding Code: GenAI should not be used to write blanket code for you with no work on your part--the goal is to train you, not it--but it can help you get started, find and fix problems and suggest improvements.

Using Remote Testing Software
☒ This course does not use remote testing software.

☐ This course uses remote testing software. Remote test-takers may choose their remote testing locations. Please note, however, that the testing software used for this may conduct a brief scan of remote test-takers' immediate surroundings, may require use of a webcam while taking an exam, may require the microphone be on while taking an exam, or may require other practices to confirm academic honesty. Test-takers therefore shall have no expectation of privacy in their test-taking location during, or immediately preceding, remote testing. If a student strongly objects to using test-taking software, the student should contact the instructor at the beginning of the semester to determine whether alternative testing arrangements are feasible. Alternatives are not guaranteed.

Required University Syllabus Statements

Accommodations/Students with Disabilities
Students needing accommodations due to a permanent or temporary disability, pregnancy or pregnancy-related conditions may contact UVU Accessibility Services at accessibilityservices@uvu.edu or 801-863-8747.

Accessibility Services is located on the Orem Campus in BA 110.

Deaf/Hard of Hearing students requesting ASL interpreters or transcribers can contact Accessibility Services to set up accommodations. Deaf/Hard of Hearing services can be contacted at DHHservices@uvu.edu

DHH is located on the Orem Campus in BA 112.

Academic Integrity
At Utah Valley University, faculty and students operate in an atmosphere of mutual trust. Maintaining an atmosphere of academic integrity allows for free exchange of ideas and enables all members of the community to achieve their highest potential. Our goal is to foster an intellectual atmosphere that produces scholars of integrity and imaginative thought. In all academic work, the ideas and contributions of others must be appropriately acknowledged and UVU students are expected to produce their own original academic work.

Faculty and students share the responsibility of ensuring the honesty and fairness of the intellectual environment at UVU. Students have a responsibility to promote academic integrity at the university by not participating in or facilitating others' participation in any act of academic dishonesty. As members of the academic community, students must become familiar with their rights and responsibilities. In each course, they are responsible for knowing the requirements and restrictions regarding research and writing, assessments, collaborative work, the use of study aids, the appropriateness of assistance, and other issues. Likewise, instructors are responsible to clearly state expectations and model best practices.

Further information on what constitutes academic dishonesty is detailed in UVU Policy 541: *Student Code of Conduct*. For the CS Department's "Code of Ethics," please see: https://www.uvu.edu/cs/ethics.html.

Equity and Title IX
Utah Valley University does not discriminate on the basis of race, color, religion, national origin, sex, sexual orientation, gender identity, gender expression, age (40 and over), disability, veteran status, pregnancy, childbirth, or pregnancy-related conditions, citizenship, genetic information, or other basis protected by applicable law, including Title IX and 34 C.F.R. Part 106, in employment, treatment, admission, access to educational programs and activities, or other University benefits or services.

Inquiries about nondiscrimination at UVU may be directed to the U.S. Department of Education's Office for Civil Rights or UVU's Title IX Coordinator at 801-863-7999 – TitleIX@uvu.edu – 800 W University Pkwy, Orem, 84058, Suite BA 203.

Religious Accommodation

UVU values and acknowledges the array of worldviews, faiths, and religions represented in our student body, and as such provides supportive accommodations for students. Religious belief or conscience broadly includes religious, non-religious, theistic, or non-theistic moral or ethical beliefs as well as participation in religious holidays, observances, or activities. Accommodations may include scheduling or due-date modifications or make-up assignments for missed class work.

To seek a religious accommodation, a student must provide written notice to the instructor and the Director of Accessibility Services at accessibilityservices@uvu.edu. If the accommodation relates to a scheduling conflict, the notice should include the date, time, and brief description of the difficulty posed by the conflict. Such requests should be made as soon as the student is aware of the prospective scheduling conflict.

While religious expression is welcome throughout campus, UVU also has a specially dedicated space for meditation, prayer, reflection, or other forms of religious expression.