CSCE 4110: Algorithms

General Information

Instructor:	Tong Shu
E-mail:	Tong.Shu@unt.edu
Office:	Discovery Park NTDP D201
Office hours:	4:00 PM - 5:00 PM on Mondays and Wednesdays
Final exam:	1:30 PM - 3:30 PM on Monday, December 8th, 2025

Course Description

Time complexity of algorithms; algorithm design methodologies including divide and conquer, greedy, and dynamic programming; exposure to approximation algorithms for NP-hard problems; performance evaluation of algorithms.

Prerequisite

Students planning to enroll in this course should have taken the course CSCE 2110 - “Foundations of Data Structures” with a C or better or CSCE 3110 - “Data Structures and Algorithms” with a C or better. Meanwhile, C/C++ programming will be helpful for this course. Students should have been exposed to the following:
(i) Time and space analysis: asymptotic notation,
(ii) Basic sorting algorithms: insertion sort, merge sort, and heap sort, and quick sort,
(iii) Data structures including trees, heaps, binary search trees, and graphs,
(iv) Recurrence relations and proof techniques,
(v) Graphs: breadth-first search, depth-first search, minimum spanning tree (Prim's and Kruskal's algorithms),
(vi) Mathematical structures: sets, relations,
(vii) Important mathematical manipulations: sums, combinatorics.

Required Textbook

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms (4th Edition). The MIT Press, 2022, ISBN: 978-0262046305

Reference Book

Michael A. Nielsen and Isaac L. Chuang. Quantum Computation and Quantum Information (10th Anniversary Edition). Cambridge University Press, 2011, ISBN: 978-1107002173

Course Syllabus

- Time complexity of recursive and randomized algorithms (Chapters 2, 3 and 5 for 2 weeks)
- Divide-and-conquer (Chapter 4 for 1 week)
- Sorting and order statistics (Chapters 7, 8 and 9 for 1 week)
- Tree-based data structures, such as red-black trees, order-statistic tree, interval trees, disjoint sets (Chapters 13, 17, and 19 for 2 weeks)

- Amortized analysis (Chapter 16 for 1 week)
- Dynamic programming (Chapter 14 for 1 week)
- Greedy algorithms (Chapter 15 for 1 week)
- Graph algorithms (Chapters 21, 22, 23 and 25 for 2 weeks)
- NP-completeness (Chapter 34 for 1 week)
- Fundamental quantum concepts: Introduction and overview (2 weeks)

This syllabus is subject to change based on the needs of the class.

Evaluation

Grading components:

Quizzes and questionnaires (individual)	10%
In-class activities	5%
Homework (individual)	30%
Project (3-student team)	10%
Mid-term exam (individual)	15%
Final exam (individual)	30%

Grading scale for undergraduate students (based on 100 points):

Grade	Score
A	90 – 100
B	80 – 89
C	70 – 79
D	60 – 69
F	59 and below

In-class activities:
Students are expected and encouraged to attend classes. Students will be responsible for any missing class activities (such as quizzes, in-class assignments, etc.) or announcements. The absence reason could be anything including university-sponsored events. Also, the student's absence does not change the due date of any assignment.

Late Policy:
Students are expected to complete work on schedule. Late work within 24 hours can only have 80% credit for delay penalty. Late work beyond 24 hours is not accepted with any excuse. For multiple submissions with partial delay, 80% credit is only applied to the delayed portion.

Academic Integrity and Student Conduct

Please refer to https://engineering.unt.edu/cse/students/resources/academic-integrity.html

Plagiarism or cheating behavior in any form is unethical and detrimental to proper education and will not be tolerated. All work submitted by a student (projects, programming assignments, lab assignments, quizzes, tests, etc.) is expected to be a student's own work. The plagiarism is incurred when any part of anybody else's work is passed as your own (no proper credit is listed to the sources in your own work) so the reader is led to believe it is therefore your own effort. Students are allowed and encouraged to discuss with each other and look up resources in the literature (including the internet) on their assignments, but appropriate references must be included for the materials consulted, and appropriate citations made when the material is taken verbatim.

This course follows the Department of Computer Science and Engineering Cheating Policy. If plagiarism or cheating occurs, the student will receive a failing grade on the assignment and (at the instructor's discretion) a failing grade in the course. Specifically, students caught cheating or plagiarizing will receive a "0" for that particular assignment or exam for the first offense. Additionally, the incident may be reported to the Dean of Students, who may impose a further penalty. The second instance of cheating in this class will result in a grade of F in the class, and referral to the Department Chairperson and Dean of Engineering, whereby a dismissal hearing may be initiated by the Dean of Engineering.
Individual assignments, including laboratory exercises and programming assignments, in this course, must be the sole work of the individual student. You should not work with other students on shared program solutions or use solutions found on the Internet. Specifically, you should never copy someone else's solution or code, and never let a classmate examine your code. If you are having trouble with an assignment, please consult with your instructor or TA/IA assigned to this course. Failure to adhere to these strict standards may be cause for disciplinary action even leading to expulsion from the University.

Students are responsible for being familiar with the university standard for academic integrity. In the case that the above description or any in-class discussion of appropriate and inappropriate collaboration does not answer all of your questions, please meet with your instructor and look at the university Student Rights and Responsibilities web page.