University of North Texas

CSCE 2110
Foundations of Data Structures

Introduction and Course Overview

Plan for Today

- Review of Course
  - Organization
  - Content

Instructor Information

- Self-introduction

Instructor: Prof. Heng Fan (please call me Heng)
Ph.D. from Stony Brook University

Email: heng.fan@unt.edu (best way to reach out)
Office: Discover Park F284
Phone: 940-565-3209

Research interest: Computer Vision, Artificial Intelligence
Website: https://hengfan2010.github.io/

Course Webpage and Schedule

- Course webpage
  - Canvas or https://hengfan2010.github.io/teaching/25F-2110/
  - Course materials, including slides, assignments, quizzes, etc.
- Schedule
  - Time: Monday & Wednesday 1:00-2:20 pm
  - Classroom: LIFE A204
- Office hours
  - Monday 8:30 - 10:30 am or by appointment
- Syllabus
  - https://hengfan2010.github.io/teaching/25F-2110/Syllabus_CSCE_2110.pdf (Check it our on your own)

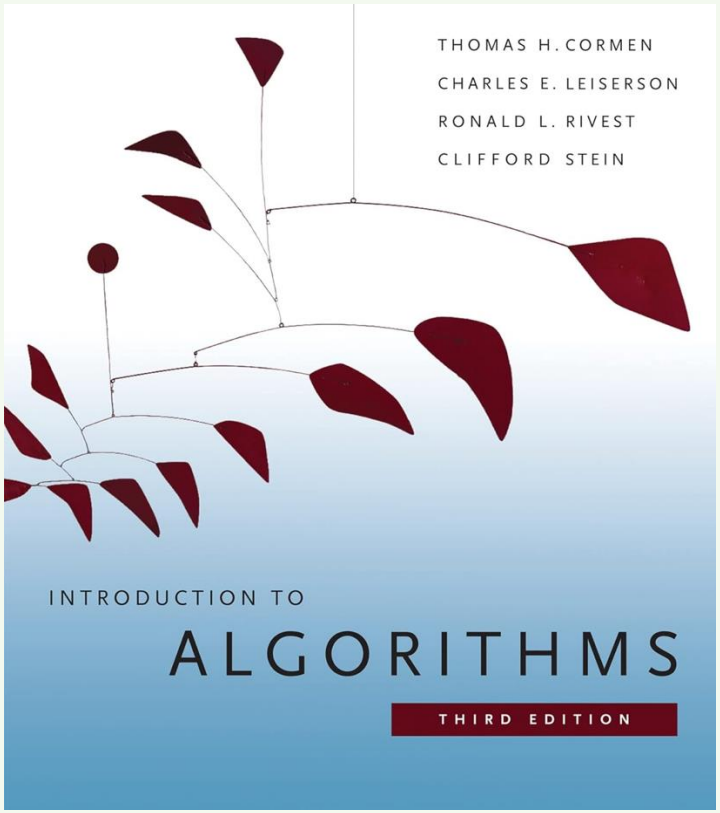
Teaching Assistant (TA)

- Bing Fan: TA for Section 211
  - Office hours: 2:00 - 4:00 pm on Monday or by appointment
  - Office: F232
  - E-mail: bingfan@my.unt.edu
- Phongsiri Nirachornkul: TA for Sections 213 & 215
  - Office hours: 11:00 - 1:00 pm on Tuesday or by appointment
  - Office: GAB 330
  - E-mail: PhongsiriNirachornkul@my.unt.edu

They will be with us to help you.

Textbooks

- Textbook
  - Introduction to Algorithms (3rd edition), by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein



computer science

Introduction to Algorithms
Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
Third Edition

Some books on algorithms are rigorous but incomplete; others cover masses of material but lack rigor. *Introduction to Algorithms* uniquely combines rigor and comprehensiveness. The book covers a broad range of algorithms in depth, yet makes their design and analysis accessible to all levels of readers. Each chapter is relatively self-contained and can be used as a unit of study. The algorithms are described in English and in a pseudocode designed to be readable by anyone who has done a little programming. The explanations have been kept elementary without sacrificing depth of coverage or mathematical rigor.

The first edition became a widely used text in universities worldwide as well as the standard reference for professionals. The second edition featured new chapters on the role of algorithms, probabilistic analysis and randomized algorithms, and linear programming. The third edition has been revised and updated throughout. It includes two completely new chapters, on van Emde Boas trees and multithreaded algorithms, and substantial additions to the chapter on recurrences (now called "Divide-and-Conquer"). It features improved treatment of dynamic programming and greedy algorithms and a new notion of edge-based flow in the material on flow networks. Many new exercises and problems have been added for this edition.

As of the third edition, this textbook is published exclusively by the MIT Press.

Thomas H. Cormen is Professor of Computer Science and former Director of the Institute for Writing and Rhetoric at Dartmouth College. Charles E. Leiserson is Professor of Computer Science and Engineering at MIT. Ronald L. Rivest is Andrew and Erna Viterbi Professor of Electrical Engineering and Computer Science at MIT. Clifford Stein is Professor of Industrial Engineering and Operations Research at Columbia University.

"In light of the explosive growth in the amount of data and the diversity of computing applications, efficient algorithms are needed now more than ever. This beautifully written, thoughtfully organized book is the definitive introductory book on the design and analysis of algorithms. The first half offers an effective method to teach and study algorithms; the second half then engages more advanced readers and curious students with compelling material on both the possibilities and the challenges in this fascinating field."
—Shang-Hua Teng, University of Southern California

"*Introduction to Algorithms*, the 'bible' of the field, is a comprehensive textbook covering the full spectrum of modern algorithms: from the fastest algorithms and data structures to polynomial-time algorithms for seemingly intractable problems, from classical algorithms in graph theory to special algorithms for string matching, computational geometry, and number theory. The revised third edition notably adds a chapter on van Emde Boas trees, one of the most useful data structures, and on multithreaded algorithms, a topic of increasing importance."
—Daniel Spielman, Department of Computer Science, Yale University

"As an educator and researcher in the field of algorithms for over two decades, I can unequivocally say that the Cormen book is the best textbook that I have ever seen on this subject. It offers an incisive, encyclopedic, and modern treatment of algorithms, and our department will continue to use it for teaching at both the graduate and undergraduate levels, as well as a reliable research reference."
—Gabriel Robins, Department of Computer Science, University of Virginia

978-0-262-03384-8

The MIT Press
Massachusetts Institute of Technology
Cambridge, Massachusetts 02142
http://mitpress.mit.edu

Content

- Topics include:
  - Review of C++ and object-oriented programming
  - Analysis of algorithms (Asymptotic notation)
  - Arrays and lists
  - Stacks and queues
  - Basic and advanced tree-based data structures
  - Tree traversal algorithms
  - Hash tables
  - Sorting algorithms
  - Graph-based data structures and algorithms (breadth-first search, depth-first search)
  - Shortest distance path and minimum spanning tree problems and algorithms

ABET Outcomes

- After completing the course satisfactorily, a student is expected to
  - Understand time complexity of algorithms
  - Understand and be able to analyze the performance of data structures for searching, including balanced trees, hash tables, and priority queues
  - Apply graphs in the context of data structures, including different representations, and analyze the usage of different data structures in the implementation of elementary graph algorithms including
    - depth-first search
    - breadth-first search
    - topological ordering
    - Prim's algorithm and Kruskal's algorithm
  - Be able to code the above-listed algorithms (using C++)

Prerequisites

- CSCE 1040 (Computer Science II) or CSCE 1045 (Computer Programming II)
  - You should be familiar with algorithms
    - loops, recursivity, C++ function, etc.
    - pseudo-code
  - You should know how to program without help in C++
    - design a program, compile, run experiments, etc.
    - debug a program

Tentative Schedule

- Introduction and Course Overview,
- Review of C++ and object-oriented programming
- Analysis of algorithms, Arrays and lists
- Stacks and queues
- Basic tree-based data structures
- Tree traversal algorithms
- Advanced tree-based data structures
- Hash tables
- Sorting algorithms
- Midterm exam
- Graph-based data structures
- Graph Algorithms
- Final review and final exam

This schedule may be updated in the future. Check Canvas or course webpage!

Grading

- Quizzes: 25%
- Assignments 40%
- Midterm exam (closed book): 15%
- Final exam (closed book): 20%
- Course project (optional): 5% (Bonus)

Grading

- (In-class) Quizzes (25%)
  - During class on TBD
  - Every two or three weeks
  - 20-30 minutes
  - May discuss solutions in office hours with instructor or TA
  - At least 4 quizzes throughout the semester
  - The quiz will be online on Canvas (bring your laptop or other devices to access Canvas)

Grading

- Assignments (40%)
  - There will be 4 or 5 homework assignments
  - Assignment is due at the end of the day (11:59 pm)
  - Written and programing exercises (C++)
  - All programming will be in C/C++ and must compile on the University's server. No credit will be given for programs that do not compile
- Important notes
  - All assignments must be turned in electrically using Canvas
  - A late penalty of 5% will be applied to all late assignments for up to 5 calendar days. Assignments that are not turned in 5 days after the due date will not be accepted
  - All holidays and weekends will be counted as calendar days

Grading

- Exams (35%)
  - Midterm exam (15%)
    - during class on TBD
    - closed book
  - Final exam (20%)
    - comprehensive
    - date and time: TBD
    - closed book

Grading

- Course project (5%)
  - Optional
  - Individual work
  - Coding task using appropriate data structures and algorithms
  - Release time TBD

Grading Scale

- Tentative grading scale (based on 100 points)

  A 90-100

  B 80-89

  C 70-79

  D 60-69

  F below 60

- No absolute grading scale; appropriate letter grade cutoffs set by instructor at the end of semester.

Recitation

- Recitation sections
  - Section 211: Wednesday 11:30 - 12:20 pm (room: NTDP F270)
  - Section 213: Thursday 9:30 - 10:20 am (room: NTDP F210)
  - Section 215: Wednesday 11:30 - 12:20 pm (room: NTDP F236)
- Attending recitation is required for all students. The recitation is to let you practice what you have learned (data structures and algorithms) in the lectures. There will be practicing examples provided in recitation to help you better understand the lectures.
- TA will be there for help (Bing Fan for section 211 and Phongsiri Nirachornkul for sections 213 and 215). E-mail them to scheduled meeting for office hours.

Academic Integrity

- Academic Integrity

Academic Integrity is defined in the UNT Policy on Student Standards for Academic Integrity. Any suspected case of Academic Dishonesty will be handled in accordance with the University Policy and procedures. Possible academic penalties range from a verbal or written admonition to a grade of F in the course. Further sanctions may apply to incidents involving major violations. You will find the policy and procedures at: https://vpaa.unt.edu/ss/integrity.

- Each topic discussed in class will have associated assignment
- Students may discuss assignment problems and approaches with each other but must write their solutions individually
- Students may not copy assignment from any source (e.g., other students, the Internet)
- No collaboration is allowed in quizzes and exams

Do NOT Cheat!

AI Course Policy

- About AI tools
  - AI tools (e.g., ChatGPT, Gemini, Grok, etc) are useful. They can be used for helping you understand the lecture content
  - AI tools are NOT allowed for quizzes and home assignments
  - AI tools are NOT allowed for exams
  - AI tools are NOT allowed for project

  - The use of AI tools for quizzes/home assignments/exams/projects will be treated as plagiarism. You will lose all points if your work is identified as AI-generated submission.

Make-up Policy

- About make-up policy for this course
  - For most situations there will be no make-up work for any assessment
  - In the event of an unavoidable absence for one of the reasons below, email the instructor as soon as possible so we can work out a solution
    - Attend a conference in which you are presenting;
    - Being in an athletic or other UNT associated event in which you are an active participant
    - A family emergency
    - A severe illness
    - Other cases
  - Provide your proof (conference registration, event registration, detailed reasons, doctor's note, etc)

Questions?