Welcome to the Course of Advanced Algorithm Design (ACS-7101/3)

Name: Yangjun Chen

PhD: University of Kaiserslautern, Germany, in 1995

Post Doctor: University of Chemnitz, Germany, 1995/7 - 1997/8

Senior engineer: Germany Research Center for Information Technology, 1997/9 - 2000/2

Post Doctor.: University of Alberta, 2000/2 - 2000/6

Assistant Prof.: University of Winnipeg, 2000/7 – 2004/6
Associate Prof.: University of Winnipeg, from 2004/7
Full Prof.: University of Winnipeg, from 2009/7

Professor: Dr. Y. Chen

Office: 3D27

home-page: http://www.acs.uwinnipeg.ca/ychen2
E-mail: y.chen@uwinnipeg.ca

phone: 204-786-9417

Meeting time: Monday and Wednesday 2:30 – 3:45 pm

Meeting location: 3D03

Office hours: 16:00 - 17:00 Monday, Wednesday
12:00 – 3:00 pm Friday (except time for dept. meeting)

Course Outline

Intro to algorithm design, analysis, and applications

Algorithm Analysis

Asymptotic Notation, Recurrence Relations, Complexity analysis, Proof Techniques.

Data Structures

Lists, Heaps, Graphs, Trees, Balanced Trees, Hash Tables.

Sorting & Ordering

Mergesort, Quicksort, Heapsort, Linear-time Sorts (bucket, counting, radix sort), Selection, Other sorting methods. Emphasis – time complexity and correctness.

Algorithmic Design Paradigms

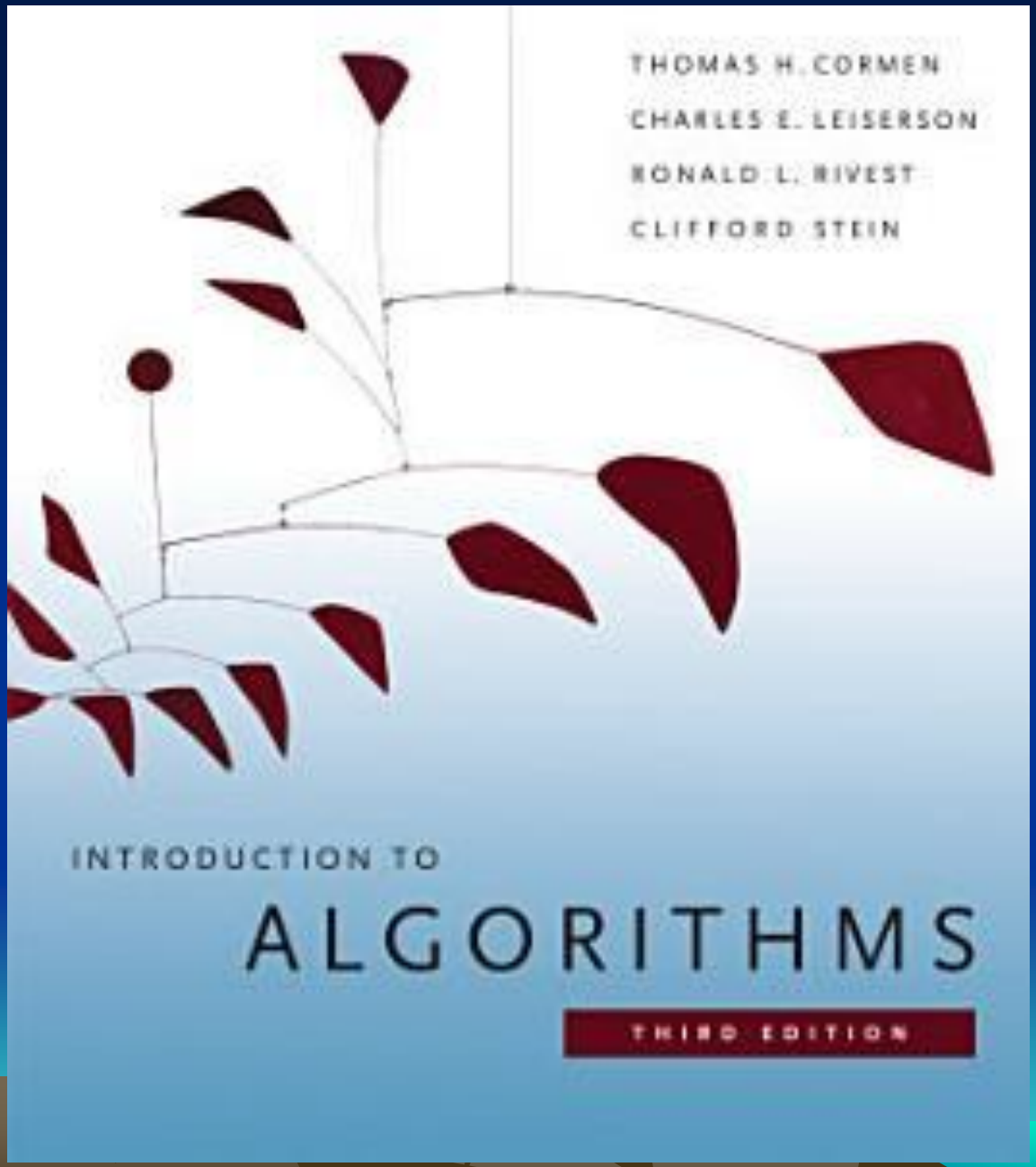
Divide and Conquer, Dynamic Programming, Greedy Algorithms, Search Trees, Graph Algorithms, String Matching, Network flow, Bipartite Graphs, Quantum Computation

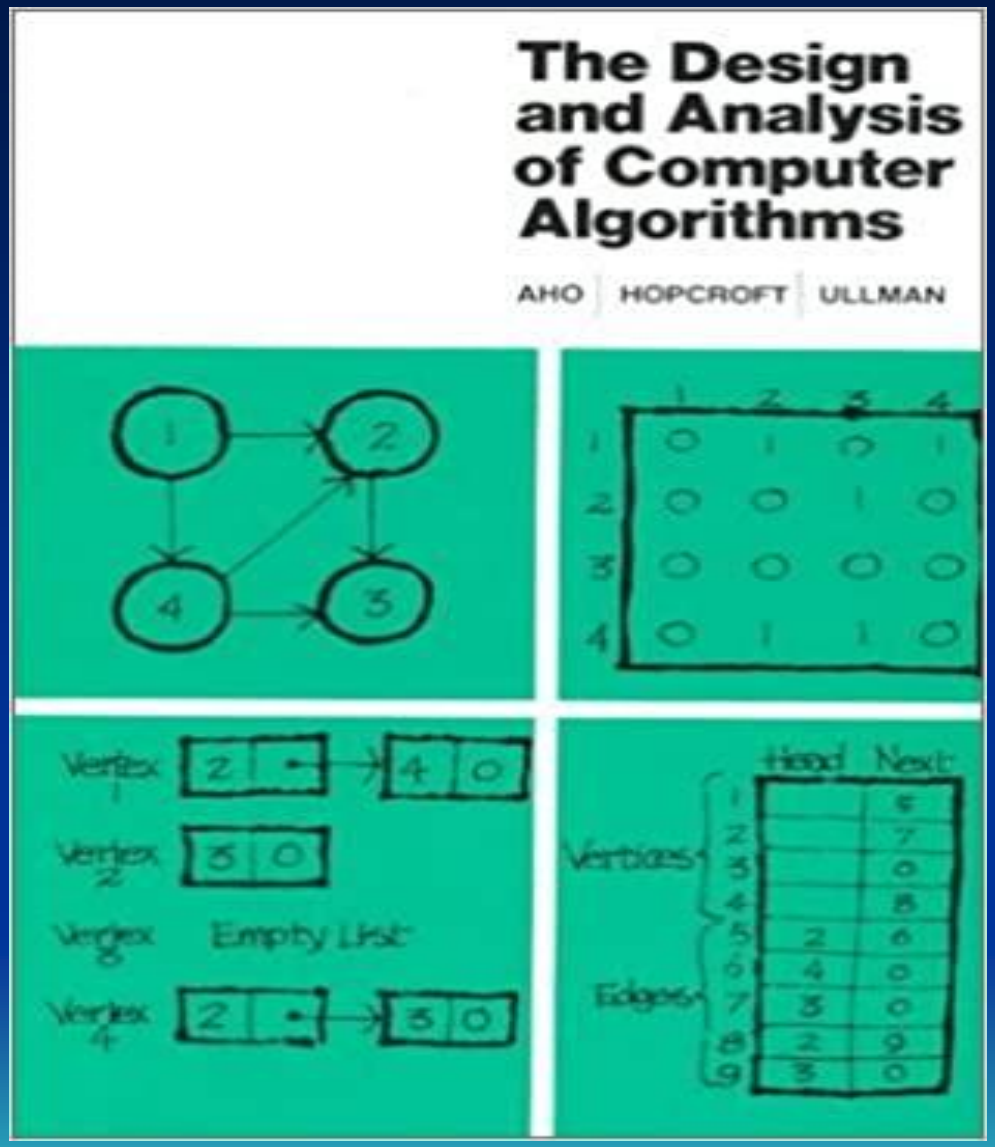
Required textbook:

- ***Introduction to Algorithms*, 2nd Ed., 3rd Ed. by Cormen, Leiserson, Rivest, & Stein (CLRS), McGraw Hill, 2002.**
- **Lecture slides online**

Other reference:

- ***The Design and Analysis of Computer Algorithms*, A.V. Aho, J.E. Hopcroft and J.D. Ullman**


THOMAS H. CORMEN
CHARLES E. LEISERSON
RONALD L. RIVEST
CLIFFORD STEIN
INTRODUCTION TO
ALGORITHMS
THIRD EDITION


The Design and Analysis of Computer Algorithms
AHO
HOPCROFT
ULLMAN
1
2
4
3
1 2 3 4
1 0 1 0 1
2 0 0 1 0
3 0 0 0 0
4 0 1 1 0
Vertex 1
2
4
0
Vertex 2
3
0
Vertex 3
Empty List
Vertex 4
2
3
0
head
Next
Vertices
Edges
1 6
2 7
3 0
4 8
5 2 6
6 4 0
7 3 0
8 2 9
9 3 0

Course Roadmap

- Algorithmics Basics
- Divide and Conquer
- Sorting and Selection
- Search Trees
- Graph Algorithms
- Dynamic Programming
- Greedy Algorithms
- Selected Topics
  - Maximum flow, String matching, Bipartite graphs
- Quantum Computation

Algorithmics Basics

- Introduction to algorithms, complexity, and proof of correctness. (Chapters 1 & 2)
- Asymptotic Notation. (Chapter 3.1)
- ***Goals***
  - *Know how to write formal problem specifications.*
  - *Know about computational models.*
  - *Know how to measure the efficiency of an algorithm.*
  - *Know the difference between upper and lower bounds of computational complexity.*
  - *Be able to prove algorithms correctness.*

Divide-and-Conquer

- Designing Algorithms using Divide-and-Conquer paradigm (Chapter 2.3)
- Recurrences (Chapter 4)
- Mergesort (Chapter 7)
- ***Goals***
  - *Know when the divide-and-conquer paradigm is an appropriate one.*
  - *Know the general structure of such algorithms.*
  - *Express their complexity using recurrence relations. Determine the complexity using techniques for solving recurrences.*
  - *Memorize the common-case solutions for recurrence relations.*

Sorting & Selection

- Quicksort (Chapter 7)
- Heapsort (Chapter 6)
- Bucket Sort, Radix Sort, etc. (Chapter 8)
- Selection (Chapter 9)
- Other Sorting Methods (Handout)
- ***Goals***
  - *Know the performance characteristics of each sorting algorithm, when they can be used, and practical coding issues.*
  - *Know the applications of binary heaps.*
  - *Know why sorting is important.*
  - *Know why linear-time median finding is useful.*

Search Trees

- Binary Search Trees – Not balanced (Chapter 12)
- Red-Black Trees – Balanced (Chapter 13)
- ***Goals***
  - *Know the characteristics of the trees.*
  - *Know the capabilities and limitations of simple binary search trees.*
  - *Know why balancing heights is important.*
  - *Know the fundamental ideas behind maintaining balance during insertions and deletions.*
  - *Be able to apply these ideas to other balanced tree data structures.*

Graph Algorithms

- Basic Graph Algorithms (Chapter 22)
- Topological sorting
- Recognizing strongly connected components
- ***Goals***
  - *Know how to represent graphs* (*adjacency matrix and edge-list representations*).
  - *Know the basic techniques for graph searching*: breadth-first searching, depth-first searching
  - *Be able to devise other algorithms based on graph-searching algorithms.*

Dynamic Programming

- Dynamic Programming (Chapter 15)
- *Find the longest common subsequences*
- *Optimal Binary Search Trees (Dictionary Construction)*
- ***Goals***
  - *What is the dynamic programming?*
  - *Know when to apply dynamic programming and how it differs from divide and conquer.*

Greedy Algorithms

- Greedy Algorithms (Chapter 16)
- Activity Selection Problem (Chapter 16)
- Minimum Spanning Trees (Chapter 23)
- ***Goals***
  - *What is a greedy algorithm?*
  - *Know when to apply greedy algorithms and their characteristics.*
  - *Be able to prove the correctness of a greedy algorithm in solving an optimization problem.*
  - *Understand where minimum spanning trees and shortest path computations arise in practice.*

Selected Topics

- Network flow
  - Ford-Fulkerson algorithm
- String matching
  - Knuth-Morris-Platt algorithm
- Bipartite graph
  - Hopcroft-Karp algorithm

Quantum Computation

- **What is a qubit?**
- **Bloch sphere interpretation**
- **About $e^{i\theta}$**
- **Qubit operators and circuits**
- **Quantum Fourier Transformation**

Randomized Algorithms

- Probability & Combinatorics. (Chapter 5)
- Quicksort. (Chapter 7)
- Hash Tables. (Chapter 11)
- ***Goals***
  - *Be able to apply the theory of probability to the following.*
    - *Design and analysis of randomized algorithms and data structures.*
    - *Average-case analysis of deterministic algorithms.*
  - *Understand the difference between average-case and worst-case runtime, esp. in sorting and hashing.*
  - *Be thorough with basic probability theory and counting theory.*

Project

- Implementing an algorithm for evaluating twig pattern queries in XML databases
  - The algorithm will be discussed in classes
- Implementing an algorithm for finding a maximum matching in a bipartite graph
  - The algorithm will be discussed in classes
- More projects will be announced

Project Report

1. Introduction (including the problem description, motivation – its significance and application in the computer engineering and industry or business)
2. Related work (describe some important techniques related to the problem to be addressed)
3. Main thrust (detailed description of the method, formal algorithm, analysis of computational complexities: time and space overhead, your design: data structure)
4. Experiments (main data structures used for implementation, description of the data used for tests, test results: charts, histogram, or tables)
5. Future work (discussion on the possible improvements, or possible extension)
6. References

Important dates:

Wednesday Sept. 04, 2024	First class
Oct. 13 – 19, 2024	reading break (no classes)
Monday, Oct. 28, 2024	Midterm examination

Nov. 13, 2024

Final date to withdraw without academic penalty from a course that begins in Sept. and ends in Dec. of the 2024 Fall term

Monday, Dec. 04, 2024	last class
Final examination	replaced by projects

Course Evaluation:

3 assignments	24%
1 midterm examination	26%
1 project (or final)	50%

- All assignments are submitted through e-mail. (Sent to the teaching assistant Ms. Rasagnya Kondam:

  kondam-r@webmail.uwinnipeg.ca)
- All works must be prepared using a word processor and placed in a folder.
- Late assignments are accepted (up to 1 day late) and receive a 25% penalty.

Academic dishonesty:

- Academic dishonesty is a very serious offense and will be dealt with in accordance with the University's discipline bylaw. Be sure that you have read and understood Regulations and Policies, #8 in the 2024-2025 UW Calendar.