Analysis of Algorithms

COMP.4040, Summer 2019

Chapter 1: Introduction & Overview

By: Sirong Lin, PhD

UMASS LOWELL

University of
Massachusetts
Lowell

Learning with Purpose

Outline

Get to Know Each Other

- start to build a learning community
- your goal in the class

Class Contents and Policy

Algorithms and Why Study

Highlights of the Course

Nature of this course

Core course: required for CS major

Advanced undergraduate level

- Graduate students take separate course(COMP.5030)

Not a programming course, more about **analysis** and **design** of algorithms

- pencil-and-paper exercises, proof
- real-world examples

Prerequisites

Computing II (COMP.1020/91.201)

Calculus I-II (MATH.1310-2310/92.131-231)

Discrete Math I & II (MATH.3201/92.321, MATH3220/92.322)

Statistics and Prob. / Statistics for Scientists and Engineers (MATH3860/92.386)

Contents

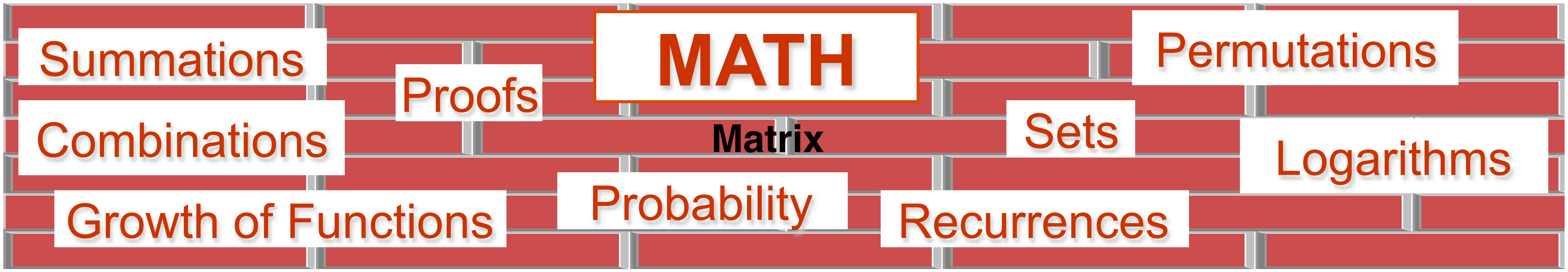
Analysis of Algorithms — Foundations

- framework to analyze & design algorithms
- growth of functions (notations)
- divide-and-conquer paradigm
- probabilistic analysis and randomized algorithms

Contents (Cont'd)

Foundations & Math Tools

- Appendix A, B1~B3, C1~C4
- Math cheat sheet on Blackboard (under Handouts)



Contents (Cont'd)

Design and Application of Algorithms

- Sorting Algorithms (e.g., heapsort, quicksort, sorting in linear time)
- Data structures (e.g., stacks, queues, linked-list, trees, hash tables) and related algorithms

Build a Learning Community — Collective Efforts

Our roles as community members

Me — Instructor /TA/Grader: facilitator, coach, mentor

- provide learning material
- offer opportunities and help
- give feedback of your learning

Our roles as community members (Cont’d)

You — active creators of knowledge, contributors of the community, responsible learners

Our roles as community members (Cont'd)

You are expected to

participate actively (don't be shy, no silly or simple questions)

show ***persistent*** efforts

provide ***appropriate*** help for your peers

give feedback of my teaching

Get to Know Each Other — why you are here

About me

Sirong Lin: Lin（林）Si (思）Rong(榕)

4th year teaching at UML, with both academic and industrial experiences

undergraduate coordinator

research interest:

Computer Science Education, Human-Computer Interaction, Computer Supported Cooperative Work, Computer Supported Collaborative Learning

Get to know each other

Find 4~5 students as a group and introduce yourself to each other

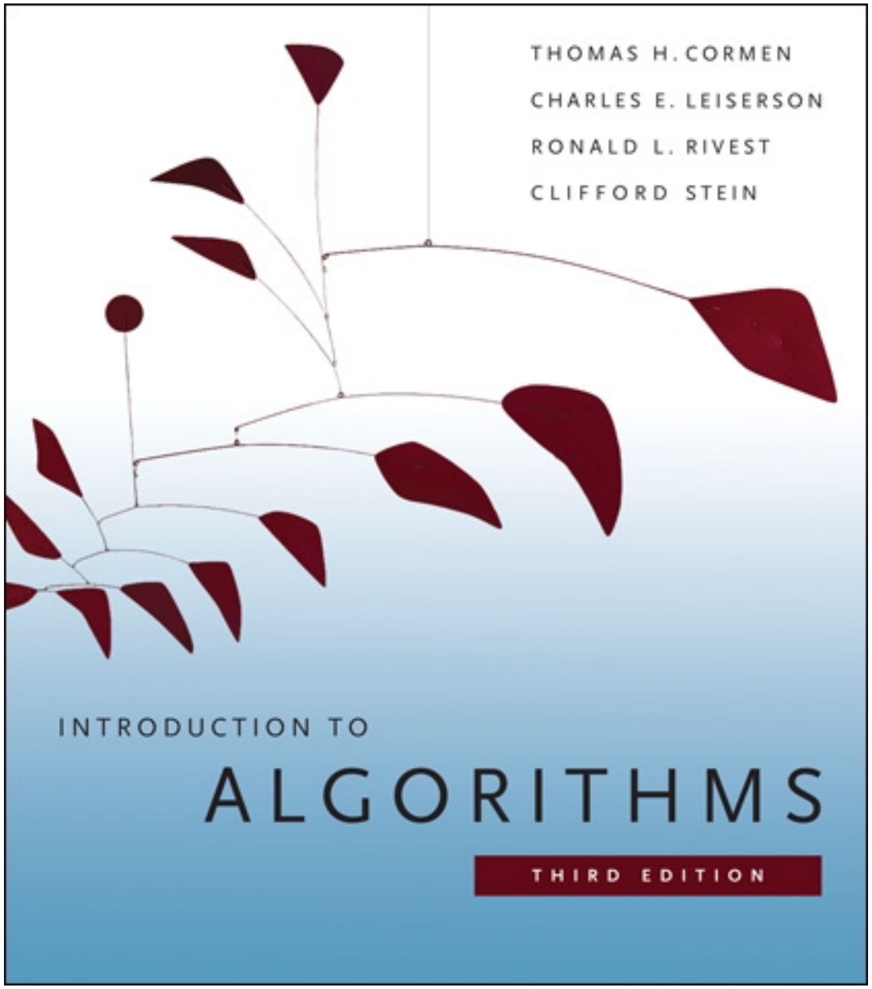
- name
- academic level, major
- your favorite algorithm
- share a learning strategy
- **what is your goal for this class?**

A representative will share something interesting about your group

More About this Class

Summer I class: 6 weeks to learn 16-week’s contents in a regular semester

Very intensive



Read textbook as we learn

Introduction to Algorithms (3rd edition)
by T.H. Cormen, C.E. Leiserson, R.L. Rivest, MIT Press,
2009. ISBN 978-0-262-03384-8

Course website

on Blackboard

- learning materials (including lecture notes, lecture videos, supplemental material, etc.)
- assignments
- your grades (e.g., assignments, quizzes)

Make Sure you have Blackboard access ASAP

Piazza — class discussion website

Enroll yourself: https://piazza.com/uml/summer2019/comp4040011

Use it to (avoid sending emails if possible):

- clarify confusion about assignments
- discuss general questions

Please don't post:

- your solutions/answers directly

How to learn

Keep thinking your learning strategy

My suggestions:

- read through the chapter
- **take notes**: important process of learning
- review notes and read textbook
- do homework and practices (e.g., redo class examples by yourself)

Assessments (grades)

Homework Average (~10): 30%

Quiz Average: 35%

Final exam (comprehensive): 30%

Discretionary (attendance, participation): 5%

for each exam, you can bring cheat sheets

Assignments

Homework is **individual** work

One problem set (one assignment) for 1~2 chapters in general

Due BEFORE the class time on the due date (in classroom)

Late assignments will not be accepted without <u>prior</u> approval (two days before the due date)

Assignments (Cont'd)

It is recommended that homework would be typed. **A paper copy must be submitted (keep a copy for yourself).**

Re-grading policy in Syllabus

- within one week after the homework is returned
- discuss the grade with the TA/grader first and then come to me

Get help outside the class

- Office hours
  - best way to contact me out of class
  - MWTh: 12:50 pm – 1:20 pm and by appointment
- keep email messages brief, put "**[4040Algorithm]**" in the title for timely reply
- Grader: Yan Lin (grade your homework): Yan_Li1@student.uml.edu

Other class policies

Formal **attendance** is mandatory. Three unexcused absences are permitted

- quizzes and class exercises may be used to take attendance
- be responsible for everything that we talk in the class if you miss the class

Refer to Syllabus

Adhere to Academic Honor Code

EXTREMELY Important!

- **Cheating, lying or plagiarizing are NOT acceptable! Serious consequences will be followed!**

You are NOT allowed to post course materials and solutions to problem sets assigned in this class in public places

Read the Syllabus and the Academic Integrity statement: http://www.uml.edu/Catalog/Undergraduate/Policies/Academic-Policies/Academic-Integrity.aspx

Homework

Read Chapter 1 & 2 (textbook)

- insertion sort, merge sort

Review Appendix A; B1, B2, B3; Appendix C1~4

- Quiz1 will test basic math skills, pro & stat, basic data structure

Some homework will not be collected, but you are still responsible for finishing them

Why Study Algorithms

Algorithms

Definition:

- a sequence of computational steps that transform the input into output
- a tool for solving a well-specified computational problem

Examples: sort numbers, quick search

Why study algorithms?

Hardware becomes faster and cheaper, still important to study algorithms?

- computing time is a bounded resource and so is space in memory
- should use them wisely and efficiently

Why study algorithms? (Cont’d)

Algorithms are core of most technologies used in Computer Science, e.g.,

- database indices rely on balanced search tree data structures
- compiler, interpreter, or assembler make extensive use of algorithms
- usability relies on fast hardware, a graphical user interface, wide-area networking

Why study algorithms? (Cont'd)

Become a better programmer

To become a world-class programmer, code 2 hours each day, for 10 years. Or

code 2 hours each day for 2 years AND take an algorithm class

FUN

Help your technical interviews!