Algorithms

Object-oriented programming

Java

Algorithms

What is an *algorithm*?

- Set of instructions for solving a problem
  - “First, wash the tomatoes.”
  - “Second, peel and cut the carrots.”
  - “Third, mix the olive oil and vinegar.“
  - “Finally, combine everything in a bowl.”
- *Finite* sequence of steps
- Unambiguous
- English, Chinese, pseudocode, Java, etc.

Algorithms

History

- Named for al-Khwārizmī (780-850)
  - Persian mathematician

- Many ancient algorithms

Algorithms

History

- Named for al-Khwārizmī (780-850)
  - Persian mathematician
- Many ancient algorithms
  - Multiplication: Rhind Papyrus
    - Babylon and Egypt: ~1800BC
  - Euclidean Algorithm: Elements
    - Greece: ~300BC
  - Sieve of Eratosthenes
    - Greece: ~200BC

“If you need your software to run twice as fast, hire better programmers.

But if you need your software to run more than twice as fast, use a better **algorithm.**”

-- *Software Lead at Microsoft*

Software: desirable features

Software

Desirable features?

- Speed / Performance
- Correctness / lack of bugs
- Memory usage
- Easy to maintain / easy to read
- Modular
- Completed on schedule
- Elegant
- Portable

Algorithms

Goals of this course:

- How to organize and manipulate data?
  - Efficiency
    - Time: *How long does it take?*
    - Space: *How much memory? How much disk?*
    - Others: Energy, parallelism, etc.
  - Scalability
    - Inputs are *large* : e.g., the internet.
    - Bigger problems consume more resources.
- Solve real (fun!) problems

Algorithms

Goals of this course:

- Discover existing “toolbox” of algorithms and data structures that you can use to solve real world problems.
- Learn how to choose the right algorithm for the right problem.
- Learn how to design and analyze new algorithms and data structures when needed.

Algorithms

How to solve a problem:

- Identify the problem
- Abstract irrelevant details
- Find good algorithms
- Implement (in Java)
- Evaluate

  How fast? How does it scale?

Semester Overview

- Topic 1: Linked data structures
  - Arrays
  - Searching
  - Sorting
  - Lists, Stacks, Queues
  - Divide-and-Conquer

- Example problems: document distance, peak finding

Semester Overview

- Topic 2: Trees
  - Binary Search Trees
  - Balanced Trees
  - Priority Queues
  - Heaps

- Example problems: simple scheduling

Semester Overview

- Topic 3: Hash Tables
  - Dictionaries
  - Hash functions
  - Chaining
  - Amortized Analysis

- Example problems: DNA similarity

Semester Overview

- Topic 4: Graphs
  - Searching in a graph
  - Spanning trees
  - Shortest paths
  - Directed graphs

- Example problems: game playing, map searching

Semester Overview

- Topic 5: Advanced Topics
  - Dynamic programming
  - Optimization
  - Numerical methods
  - Concurrent algorithms

Semester Overview

For each algorithm:

1. What problem does it solve?
2. Why does it work?
3. How do you implement it?
4. What is its asymptotic performance?
5. What is its real world performance?
6. What are the trade-offs?

Types of knowledge

1. Basic, permanent facts
2. General concepts
3. Background culture

Types of knowledge

1. Surface memorization

2. Analytic manipulation

3. Deep understanding

Java

Language Does Not Matter

Algorithms are more important:

- Fact: C can be 20x as fast as Python!

(Source: MIT 6.006, Spring 2008)

- Fast sorting in Python (merge-sort):
  - Time: $T(n) = 2n\log(n)\ \mu s$
  - Total time for 10,000 elements: 0.266s

- Slow sorting in C (insertion-sort):
  - Time: $T(n) = 0.01n^2$
  - Total time for 10,000 elements: 1s

Computer languages

- Hardware: Assembly language
- Procedural (imperative) languages:
  - Fortran, COBOL, BASIC, Pascal , C
- Functional languages:
  - IPL, Lisp, Scheme, Haskell
- Declarative languages:
  - SQL, Lex/Yacc
- Scripting languages:
  - Javascript, Python
- Object-oriented languages:
  - Simula 67, Smalltalk 80, C++, Java, C#

Objected-oriented programming

Why Java?

- Good aspects:
  - Common in industry / real-world / web
  - Modularity / Abstraction via OOP
  - Avoids memory leak issues of C/C++

- Less good aspects:
  - Performance?? (compare to: C++)
  - Elegance?? (compare to: Scheme)

How to learn Java / OOP?

Knowledge

Experience

Talent

Problem Sets

Practice, practice, practice…

Problem Sets

If the tutor does not understand your solution, then it will not be graded.

Style and comments matter.

The tutor may ask you to explain your code/ algorithm better.

The tutor is not a compiler.

Problem Sets

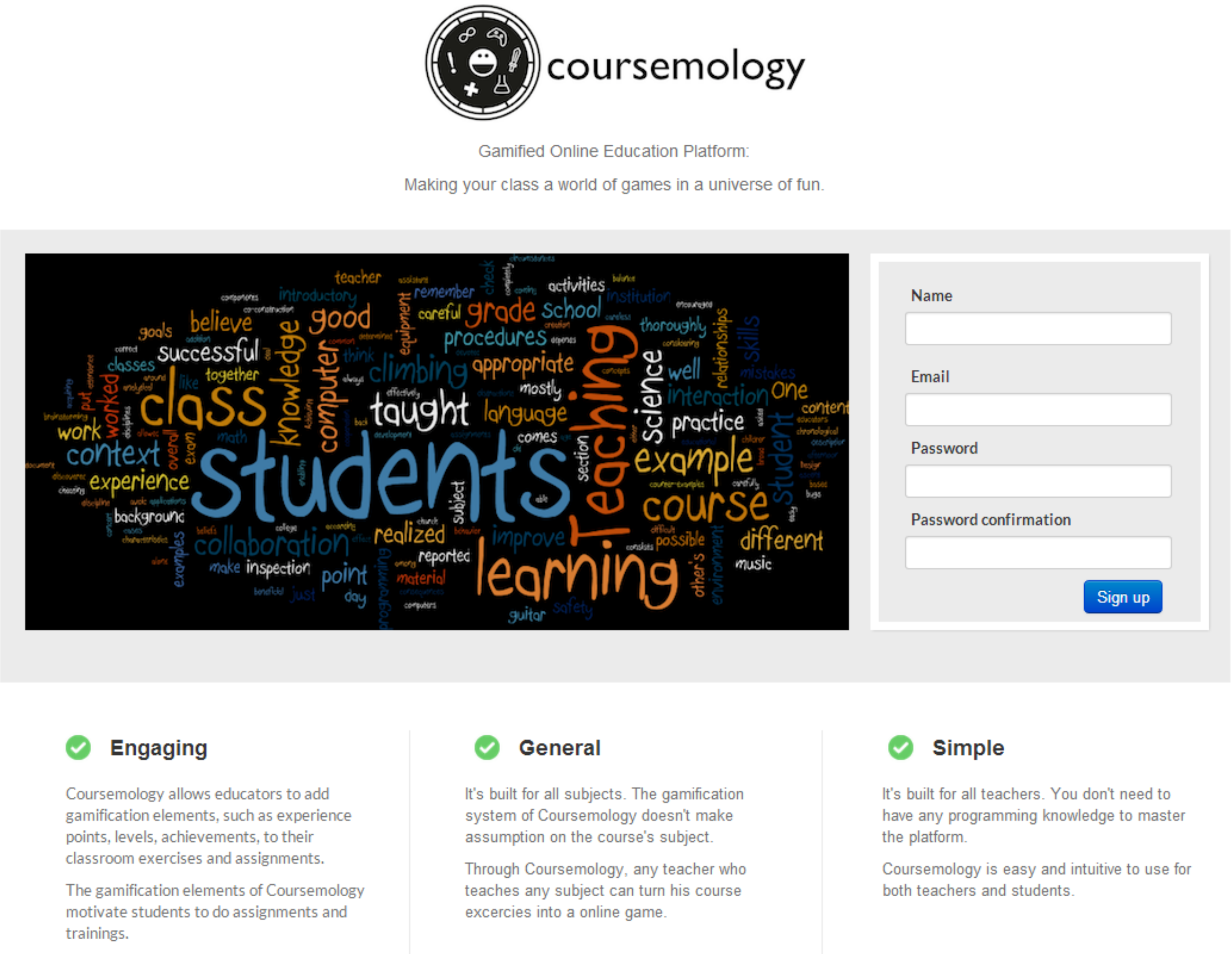
Late submissions:

- 24 hours: 20% penalty
- 2 weeks: 40% penalty
- Last day of class: 60% penalty

Hand in problem sets on time!

Even if late, do them anyways!

Coursemology



coursemology

Gamified Online Education Platform:

Making your class a world of games in a universe of fun.

Name

Email

Password

Password confirmation

Engaging

Coursemology allows educators to add gamification elements, such as experience points, levels, achievements, to their classroom exercises and assignments.

The gamification elements of Coursemology motivate students to do assignments and trainings.

General

It's built for all subjects. The gamification system of Coursemology doesn't make assumption on the course's subject.

Through Coursemology, any teacher who teaches any subject can turn his course excercies into a online game.

Simple

It's built for all teachers. You don't need to have any programming knowledge to master the platform.

Coursemology is easy and intuitive to use for both teachers and students.

Coursemology

Problem sets:

- Access problem sets
- Submit solutions
  - Upload solution as .java files.
  - Ignore code box.
- Interact with tutors
- Get grades

How to join Coursemology?

- You will receive an invitation e-mail.

Weekly Exercises

Easier problems:

- Coding challenge problems
- Designed to test basic programming skills
- Related to material from that week

Grading:

- 0/1 basis: done or not done
- Due: one week from release date

(Computer) Chess

Computer chess reels from 'biggest sporting scandal since Ben Johnson'

Czech mate, Mr Cheat

By **Tariq Tahir**

IT'S a story that has sent pawns and rooks spinning off chess boards across the world.

Rybka, the best chess-playing computer on the planet, is a cheat.

And its developer, Vasik Rajlich – one half of a couple dubbed the Posh and Becks of the game – has been shamed as a plagiarist, banned from competing, stripped of his titles and ordered to hand back his trophies and prize money.

Rajlich, himself an international master of the game, was found guilty by his peers of basically copying earlier chess programs when creating Rybka.

A 34-member panel found the 40-year-old Czech from Ohio, but now living in Hungary, plagiarised two other programs, Crafty and Fruit. Their report states: 'Not a single panel member believed him innocent. Vasik Rajlich's claims of complete originality are contrary to the facts.'

Not since IBM's Deep Blue computer defeated grand master Garry Kasparov in 1997 – and was subsequently accused of cheating – has the world of computer chess been in such uproar. Rybka won the International Computer Games Association world championship from 2007 to 2010.

Peter Doggers, from online site Chess Vibes, said: 'The impact in the computer chess world must be comparable to arguably the most famous example of doping in athletics – the positive drug testing of Canadian sprinter Ben Johnson.'

For his part, Rajlich has not commented, save an email sent to the association in which he disputed Rybka included code written by others.

He never made grand master as a player so turned to programming. 'I figured there were about 2,000 people in the world stronger than me in chess,' he once said, 'but not one chess player that was stronger than me in programming.'

By 2005, Rybka – Czech for 'little fish' – was ready. The chief tester is his wife, Iweta, herself an international master. David Levy, president of the ICGA, said: 'We are convinced the evidence against Rajlich is both overwhelming in its volume and beyond reasonable question in its nature.'

Posh and checks: Vasik Rajlich and his chief tester, wife Iweta

Problem Sets

Collaboration Policy

- Working together is strongly encouraged!

- You must write-up your problems sets alone.

- You must list on your submission the name of everyone you worked with, and all sources used.

- Cheating / plagiarism will be dealt with harshly.

Administrative Details

Weekly schedule:

- Two lectures: Wed/Fri 10am-12pm
- One recitation (1hr): Friday
- One discussion group (2hr): Mon/Tues/Thurs

Lecture slides:

- Posted after lecture (see later)
- Screencast
- Typos, typos, typos

Administrative Details

Weekly work:

- Problem set
- Discussion group problems
- Occasional competitions
- Occasional bonus problems

Administrative Details

Discussion Groups:

- Register via CORS (known as: labs)
- Fill out preference form on IVLE.
- Start in Week 3

Discussion Groups

Administrative Details

Eight Discussion Groups (tutorials):

- Monday:
  - 1-3pm
- Tuesday:
  - 10-12pm
  - 12-2pm
  - 12-2pm
  - 4-6pm
- Thursday
  - 2-4pm
  - 2-4pm
  - 4-6pm

Register on CORS…

DGs start Week 3.

Administrative Details

Problem Sessions (recitations):

- Register via CORS
- Three slots:
  - 1pm
  - 2pm
  - 3pm
- Start this week!
- Choose any of the 3 slots, temporarily!

Where to ask questions?

Discussion Forums

Facebook:

- Announcements
- Interesting related information
- Short questions

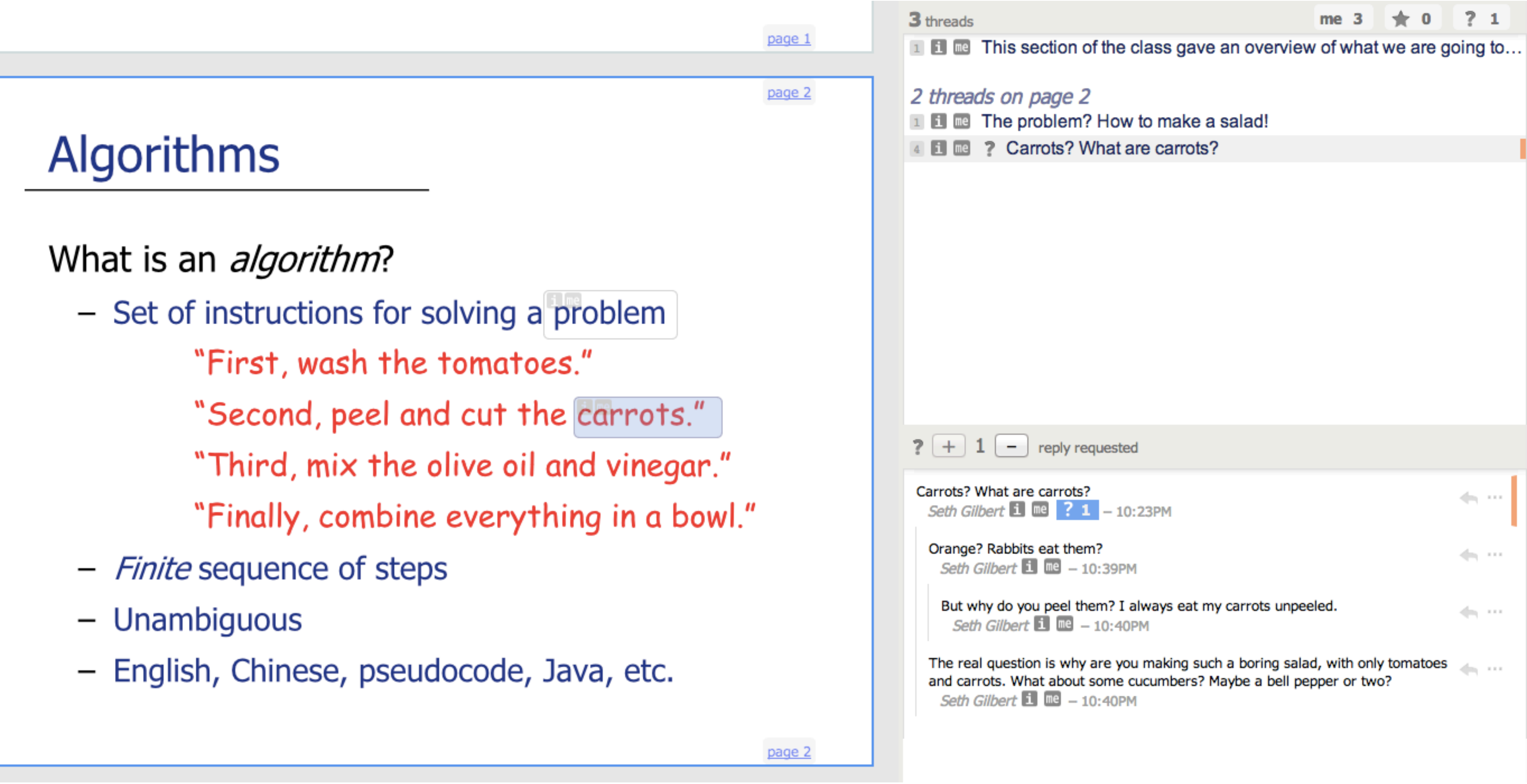
https://www.facebook.com/groups/cs2020.2014/

Discussion Forums

Nota Bena (NB):

- Annotated lecture notes
- Ongoing conversation
  - Ask questions about lecture
  - Ask questions about the problem set
- I will e-mail you link to join

http://nb.mit.edu/



page 1

page 2

Algorithms

What is an *algorithm*?

- Set of instructions for solving a problem
  - "First, wash the tomatoes."
  - "Second, peel and cut the carrots."
  - "Third, mix the olive oil and vinegar."
  - "Finally, combine everything in a bowl."
- *Finite* sequence of steps
- Unambiguous
- English, Chinese, pseudocode, Java, etc.

page 2

3 threads me 3 ★ 0 ? 1

1 i me This section of the class gave an overview of what we are going to…

2 threads on page 2

1 i me The problem? How to make a salad!

4 i me ? Carrots? What are carrots?

? + 1 − reply requested

Carrots? What are carrots?
Seth Gilbert i me ? 1 – 10:23PM

Orange? Rabbits eat them?
Seth Gilbert i me – 10:39PM

But why do you peel them? I always eat my carrots unpeeled.
Seth Gilbert i me – 10:40PM

The real question is why are you making such a boring salad, with only tomatoes and carrots. What about some cucumbers? Maybe a bell pepper or two?
Seth Gilbert i me – 10:40PM

Administrative Details

Announcements:

- E-mail: your official NUS e-mail address
- Facebook group (cs2020 2013)
- Coursemology announcements

Please check at least one of these regularly!

Administrative Details

Quizzes:

- Quiz 1 – Feb. 14
- Coding Quiz – Mar. 13
- Quiz 2 – Friday Apr. 3

(Final) Exam: Reading Week

- April 25
- Exam will be returned by the end of semester

Participation in DG / Problem Sessions

Administrative Details

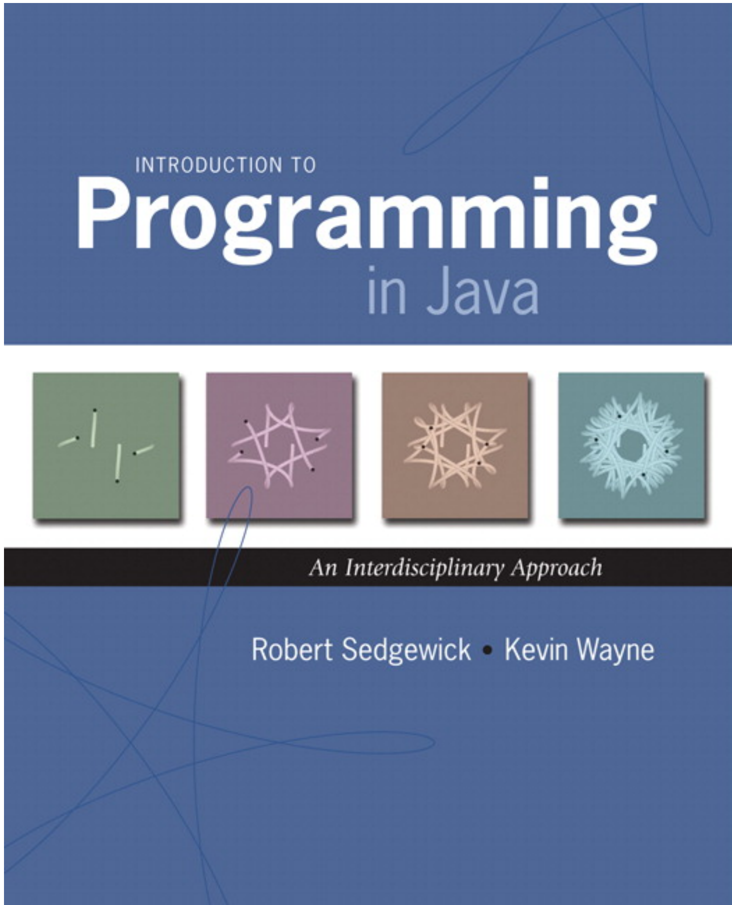
Textbook: Algorithms

- Robert Sedgewick and Kevin Wayne

Administrative Details

Textbook: Introduction to Programming in Java

- Robert Sedgewick and Kevin Wayne

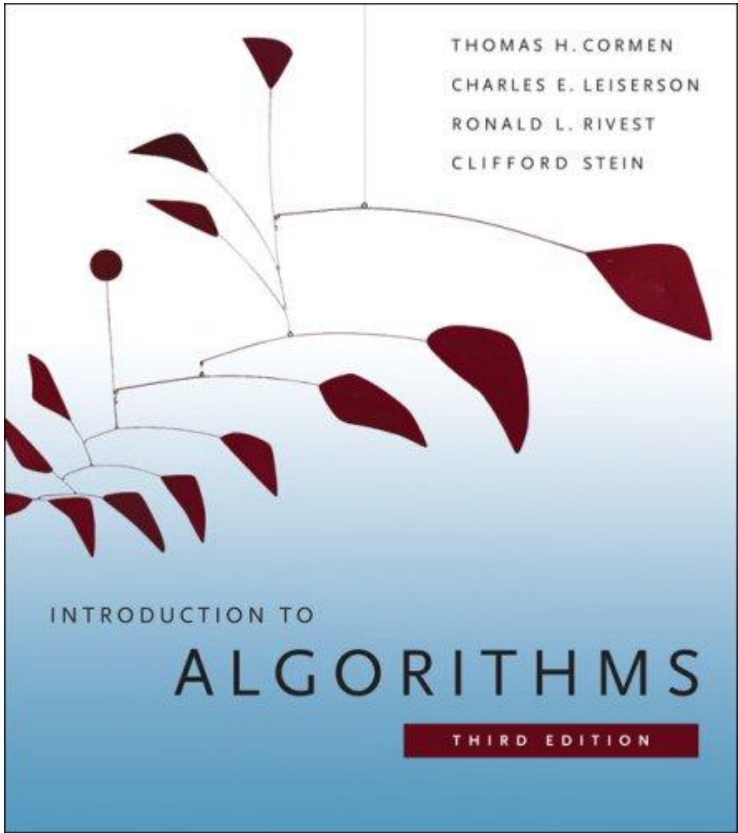


- Optional

Administrative Details

Textbook: Introduction to Algorithms

- Cormen, Leiserson, Rivest, Stein
- Optional (but recommended)



Should I take CS2020?

Administrative Details

Deadlines:

- Add module: end of Week 1 (Jan. 17)
- Drop module: end of Week 2 (Jan. 24)
- Drop module with “W”: end of recess week

Note: cannot add CS1020 after Jan. 18!

Problem Sets

"Give an algorithm…"

"Implement an algorithm…"

Five parts to each answer:

1. State the problem being solved.
2. Describe the solution in words. (If it is a Java algorithm, describe the algorithm being implemented.)
3. Give the algorithm/Java.
4. Explain why it works.
5. (Optional) Analyze its performance.