COS 226, Spring 2020

Algorithms

and

Data Structures

Kevin Wayne · Maia Ginsburg · Ibrahim Albluwi

PRINCETON
UNIVERSITY

https://algs4.cs.princeton.edu

Intro to COS 226

- *motivation*
- *course structure*
- *assessments*
- *resources*

COS 226 course overview

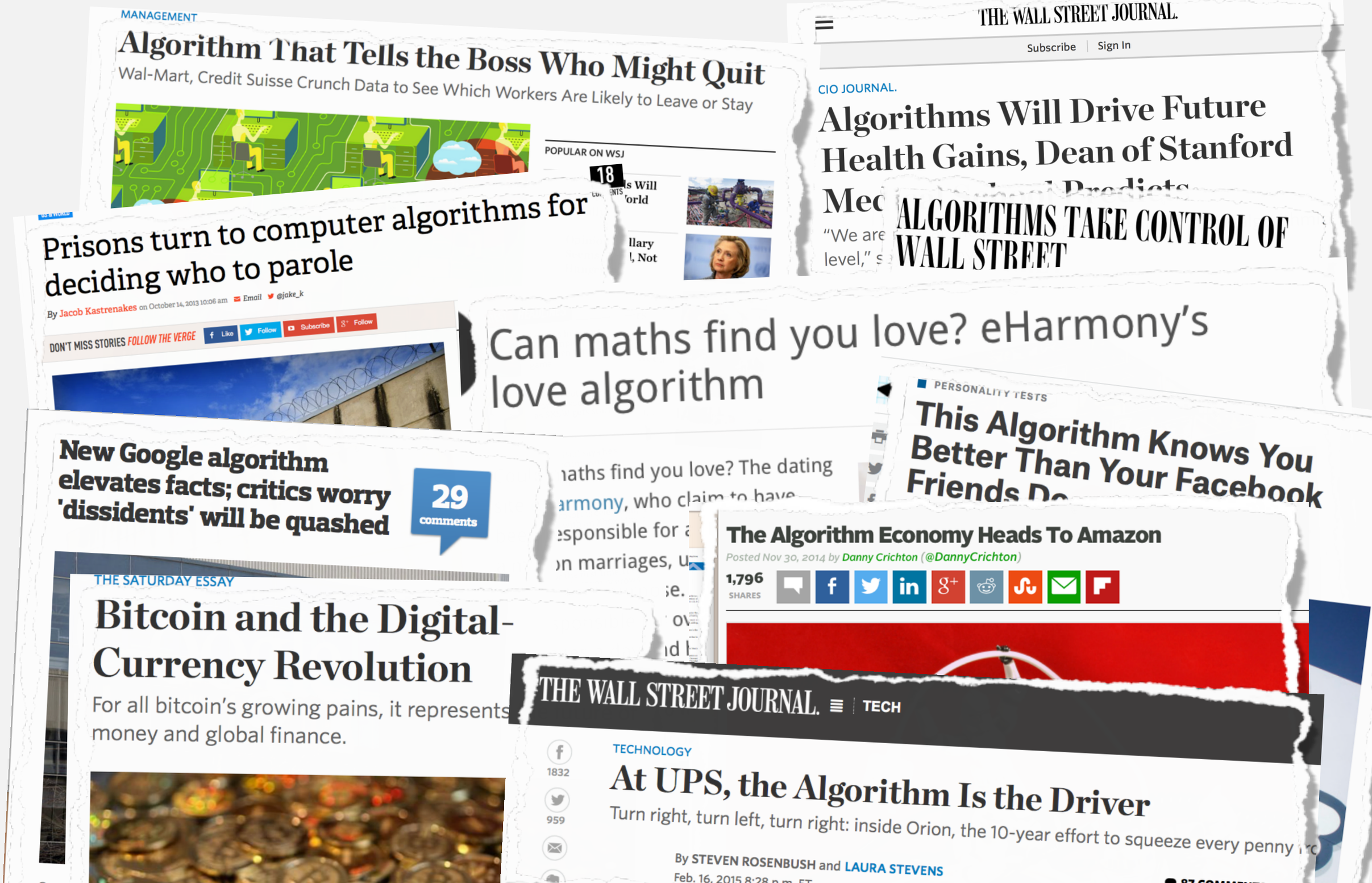
What is COS 226?

- Intermediate-level survey course.
- Programming and problem solving, with applications.
- Algorithm: sequence of instructions for solving a problem.
- Data structure: method to organize data in a computer.

topic	data structures and algorithms
data types	stack, queue, union–find, priority queue
sorting	quicksort, mergesort, heapsort, radix sorts
searching	BST, red–black BST, hash table, kd-tree
graphs	BFS, DFS, Prim, Kruskal, Dijkstra, maxflow
strings	tries, KMP, regexp, suffix arrays, data compression

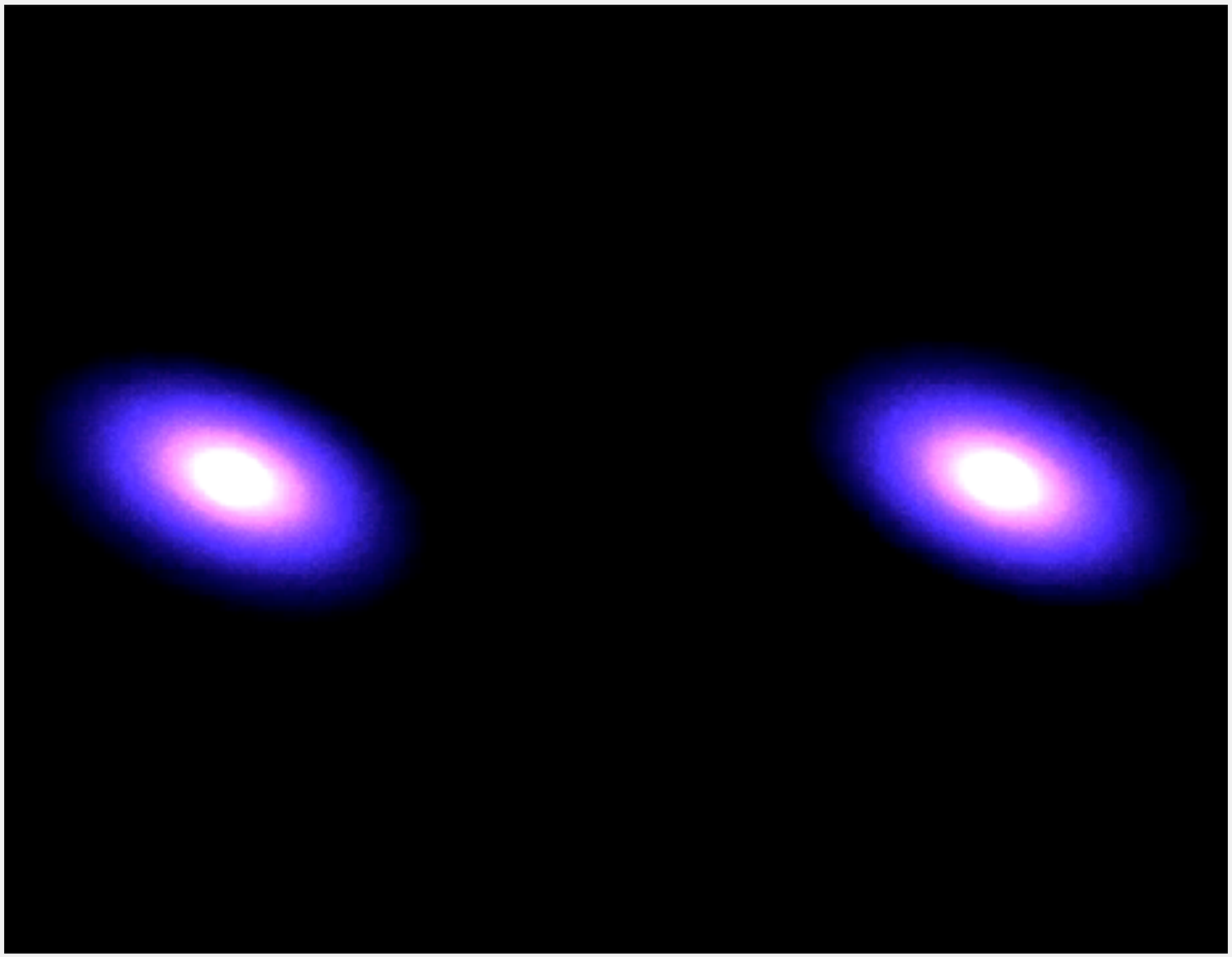
Why study algorithms and data structures?

Their impact is broad and far-reaching.



Why study algorithms and data structures?

They may unlock the secrets of life and of the universe.

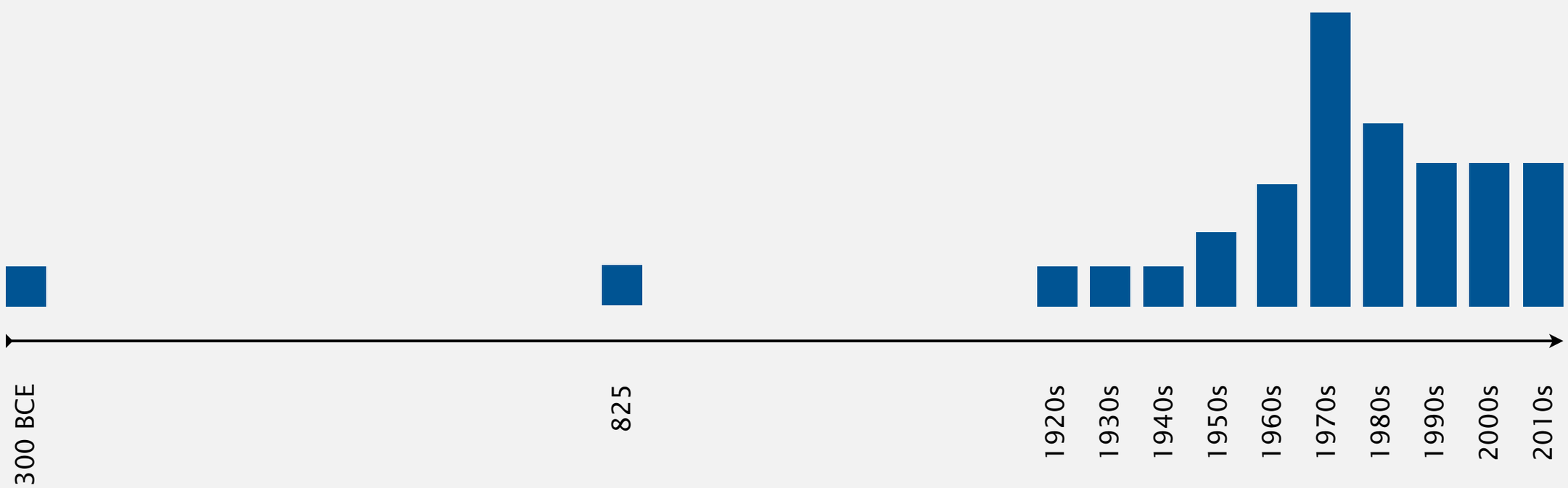


http://www.youtube.com/watch?v=ua7YlN4eL_w

Why study algorithms and data structures?

Old roots, new opportunities.

- Study of algorithms dates at least to Euclid.
- Named after Muḥammad ibn Mūsā al-Khwārizmī.
- Formalized by Church and Turing in 1930s.
- Some important algorithms were discovered
  by undergrads in a course like this!



Why study algorithms and data structures?

To become a proficient programmer.

" I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships. "

— Linus Torvalds (architect of Linux and git)

Why study algorithms and data structures?

For intellectual stimulation.

> *" For me, great algorithms are the poetry of computation. Just like verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing."* — *Francis Sullivan*

FROM THE EDITORS

THE JOY OF ALGORITHMS

Francis Sullivan, Associate Editor-in-Chief

THE THEME OF THIS FIRST-OF-THE-CENTURY ISSUE OF *COMPUTING IN SCIENCE & ENGINEERING* IS ALGORITHMS. IN FACT, WE WERE BOLD ENOUGH—AND PERHAPS FOOLISH ENOUGH—TO CALL THE 10 EXAMPLES WE'VE SELECTED "THE TOP 10 ALGORITHMS OF THE CENTURY."

Computational algorithms are probably as old as civilization. Sumerian cuneiform, one of the most ancient written records, consists partly of algorithm descriptions for reckoning in base 60. And I suppose we could claim that the Druid algorithm for estimating the start of summer is embodied in Stonehenge. (That's really hard hardware!)

Like so many other things that technology affects, algorithms have advanced in startling and unexpected ways in the 20th century—at least it looks that way to us now. The algorithms we chose for this issue have been essential for progress in communications, health care, manufacturing, economics, weather prediction, defense, and fundamental science. Conversely, progress in these areas has stimulated the search for ever-better algorithms. I recall one late-night bull session on the Maryland Shore when someone asked, "Who first ate a crab? After all, they don't look very appetizing." After the usual speculations about the observed behavior of sea gulls, someone gave what must be the right answer—namely, "A very hungry person first ate a crab."

The flip side to "necessity is the mother of invention" is "invention creates its own necessity." Our need for powerful machines always exceeds their availability. Each significant computation brings insights that suggest the next, usually much larger, computation to be done. New algorithms are an attempt to bridge the gap between the demand for cycles and the available supply of them. We've become accustomed to gaining the Moore's Law factor of two every 18 months. In effect, Moore's Law changes the constant in front of the estimate of running time as a function of problem size. Important new algorithms do not come along every 1.5 years, but when they do, they can change the exponent of the complexity!

For me, great algorithms are the poetry of computation. Just like verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing. Researchers have cracked many hard problems since 1 January 1900, but we are passing some even harder ones on to the next century. In spite of a lot of good work, the question of how to extract information from extremely large masses of data is still almost untouched. There are still very big challenges coming from more "traditional" tasks, too. For example, we need efficient methods to tell when the result of a large floating-point calculation is likely to be correct. Think of the way that check sums function. The added computational cost is very small, but the added confidence in the answer is large. Is there an analog for things such as huge, multidisciplinary optimizations? At an even deeper level is the issue of reasonable methods for solving specific cases of "impossible" problems. Instances of NP-complete problems crop up in attempting to answer many practical questions. Are there efficient ways to attack them?

I suspect that in the 21st century, things will be ripe for another revolution in our understanding of the foundations of computational theory. Questions already arising from quantum computing and problems associated with the generation of random numbers seem to require that we somehow tie together theories of computing, logic, and the nature of the physical world.

The new century is not going to be very restful for us, but it is not going to be dull either!

Why study algorithms and data structures?

For fun and profit.

Why study algorithms and data structures?

- Their impact is broad and far-reaching.
- They may unlock the secrets of life and of the universe.
- Old roots, new opportunities.
- To become a proficient programmer.
- For intellectual stimulation.
- For fun and profit.

https://algs4.cs.princeton.edu

Intro to COS 226

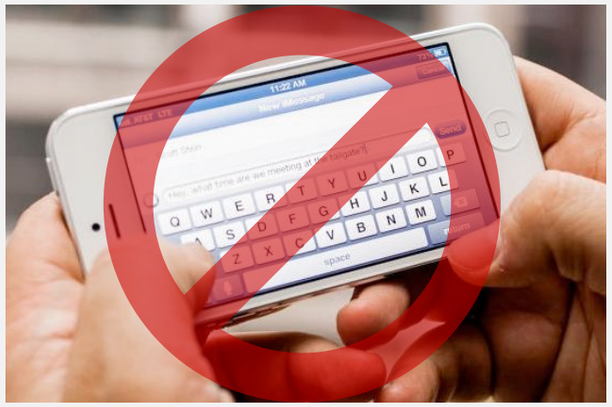
- *motivation*
- *course structure*
- *assessments*
- *resources*

Lectures

Live lectures. Introduce new material.

What	When	Where	Who	Office Hours
L01	TTh 11–12:20	Friend 101	Kevin Wayne	TBA

Electronic devices. Permitted *only* to support lecture (e.g., viewing slides and taking notes).



Student response system (required).

- Register hardware iClicker in Blackboard.
- Register iClicker Reef using Princeton email address.
- Available at Labyrinth Books ($30).

use only one device per lecture!

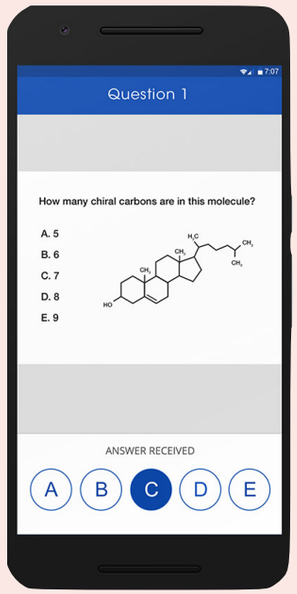
Which iClicker are you using?

A. iClicker.

B. iClicker+.

C. iClicker 2.

D. iClicker Reef.



Precepts

Discussion, problem-solving, assignment prep, …

Maia Ginsburg

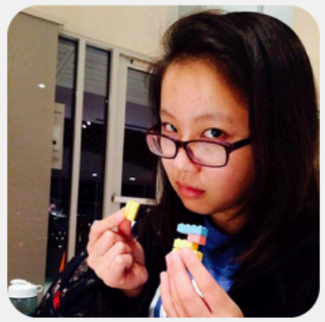
Faculty

Lead Preceptor

Ibrahim Albluwi

Faculty

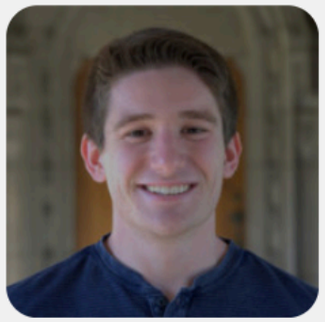
Lead Preceptor



Lisa Jian

Graduate Student

Preceptor



Chris Sciavolino

Graduate Student

Preceptor

Molly Pan

Graduate Student

Preceptor

Tim Alberdingk Thijm

Graduate Student

Preceptor

Precepts

Discussion, problem-solving, assignment prep, ...

What	When	Where	Who	Office Hours
P01	**Th 1:30–2:50pm**	**1976 Hall 028**	**TBA**	see web
P02	**Th 3–4:20pm**	**1976 Hall 028**	**TBA**	see web
P03	**F 11–12:20pm**	**Friend 009**	**TBA**	see web
P03A	**F 11–12:20pm**	**1976 Hall 028**	**TBA**	see web
P04	**F 1:30–2:50pm**	**Friend 009**	**TBA**	see web
P04A	**F 1:30–2:50pm**	**Equad E225**	**TBA**	see web
P05	**F 3–4:20pm**	**Friend 009**	**TBA**	see web
P07	**F 11–12:20pm**	**Friend 016**	**TBA**	see web

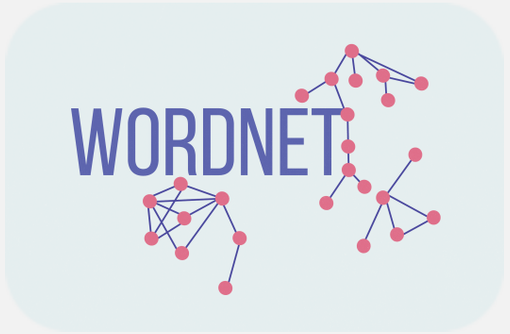
https://algs4.cs.princeton.edu

Intro to COS 226

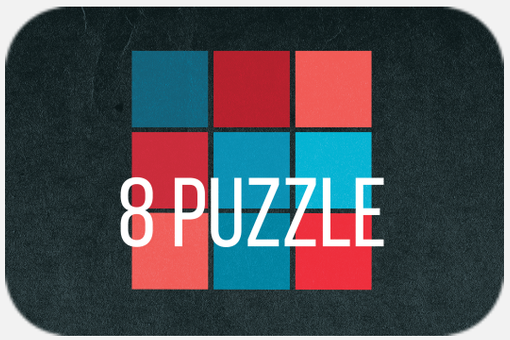
- *motivation*
- *course structure*
- *assessments*
- *resources*
- *union–find*

Programming assignments

Implement an efficient algorithm or data structure.



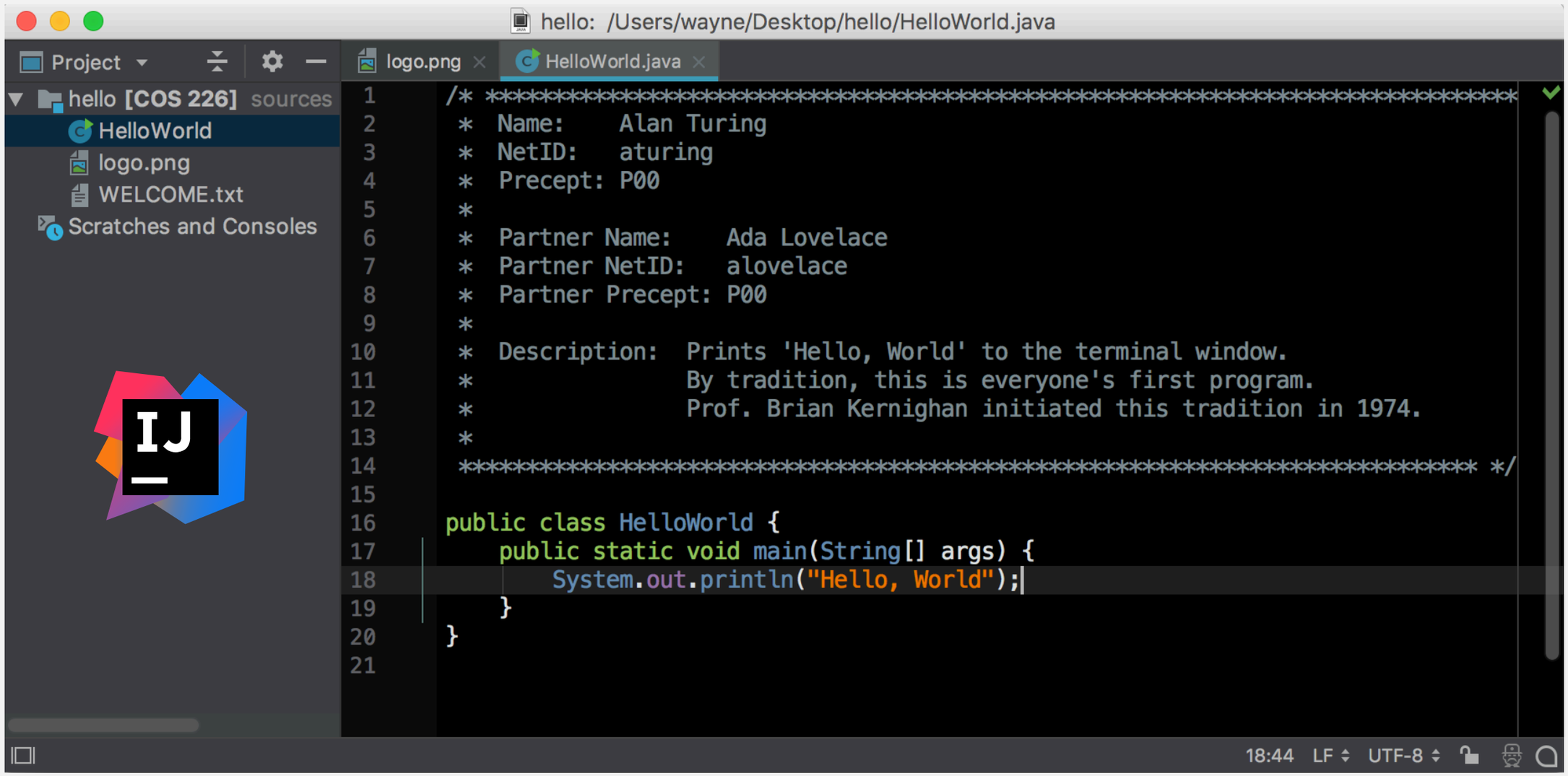
Solve an interesting application using a “textbook” algorithm.



Programming environment

Recommended IDE. Custom IntelliJ environment (used in COS 126).

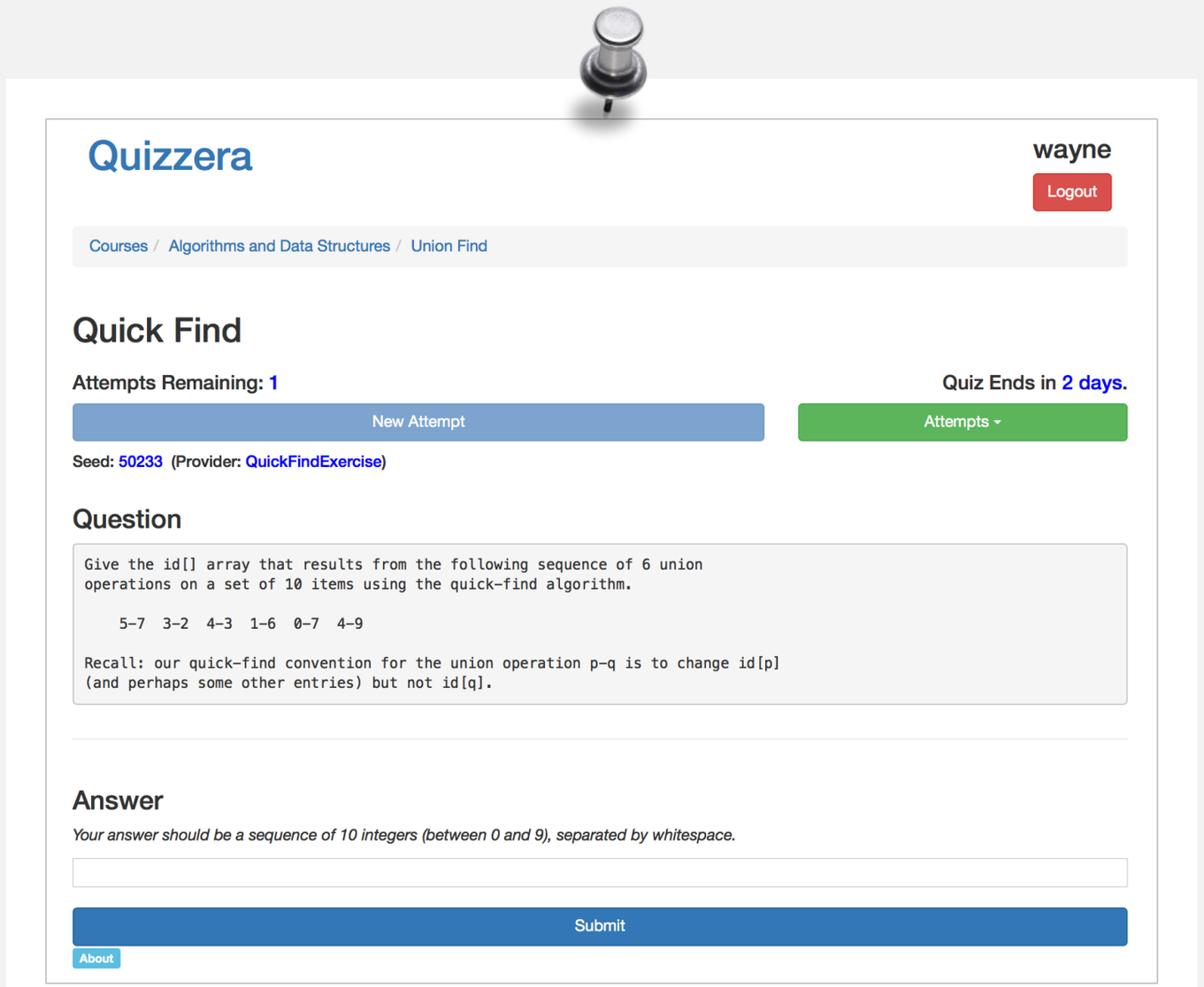
- Continuous code inspection; integrated Checkstyle and Spotbugs.
- Autoformat, autoimport, and autocomplete.
- Embedded bash terminal.
- ...



Quizzes

Quizzera platform.

- 2–3 short questions per lecture.
- Solving using pencil and paper.
- 3 attempts per question (score = max).



Midterm and final

Written exams.

- Questions drawn from quizzes and lectures.
- Emphasizes non-programming material.

COS 226 **Algorithms and Data Structures** **Fall 2017**

Midterm

This exam has 10 questions (including question 0) worth a total of 55 points. You have 80 minutes. This exam is preprocessed by a computer, so please **write darkly** and **write your answers inside the designated spaces.**

Policies. The exam is closed book, except that you are allowed to use a one page cheatsheet (8.5-by-11 paper, one side, in your own handwriting). No electronic devices are permitted.

Grading

Programming assignments. 45%

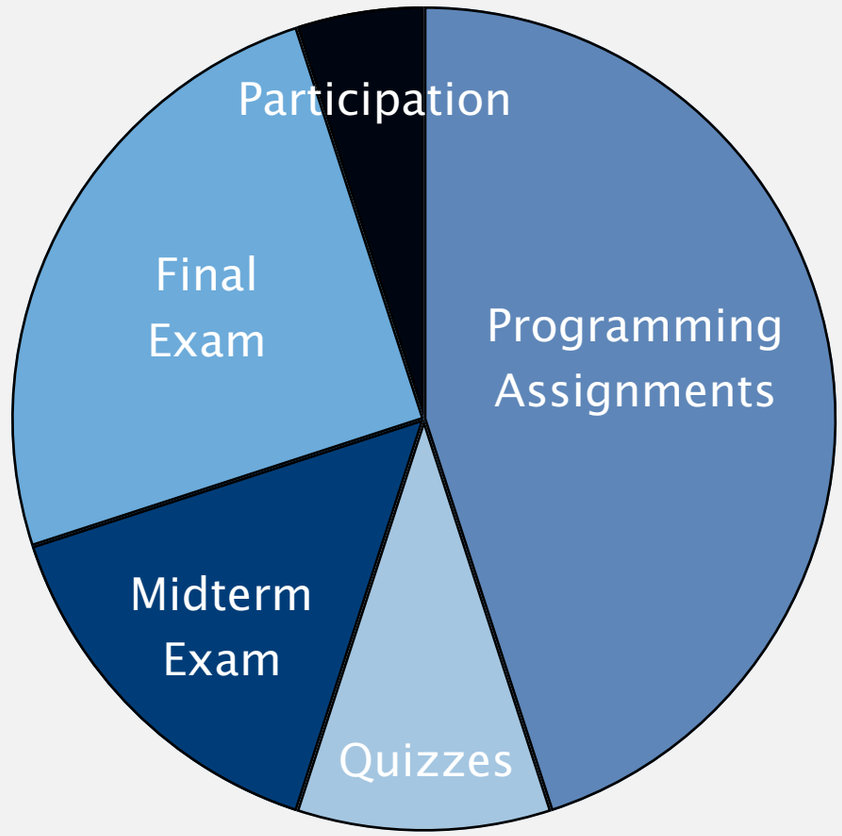
- Due at 11pm on Mondays via TigerFile.
- Collaboration/lateness policies: see web.

Quizzes. 10%

- Due at 11pm on Fridays via Quizzera.
- Collaboration/lateness policies: see web.

Exams. 15% + 25%

- Midterm (in class on Tuesday, March 24).
- Final (to be scheduled by Registrar).



Active participation. 5%

- Participate in precept/lecture.

  [perfect attendance not required to earn 100% of participation points]

- Answer questions in online discussion forum.

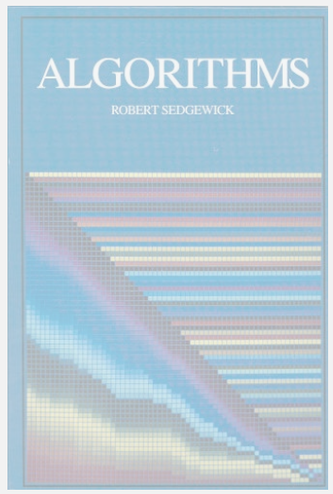
https://algs4.cs.princeton.edu

Intro to COS 226

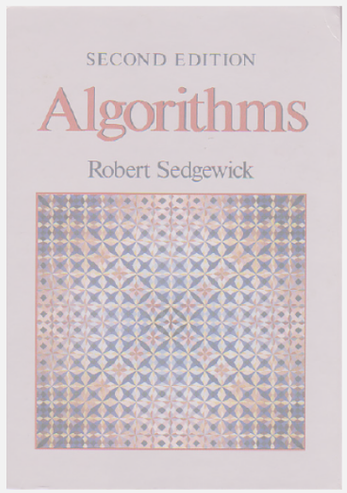
- *motivation*
- *course structure*
- *assessments*
- *resources*

Resources (textbook)

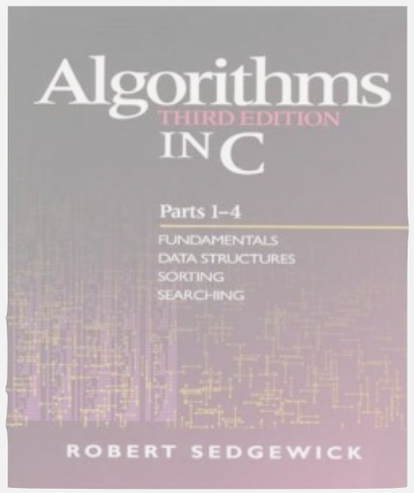
Readings (required). Algorithms 4th edition by R. Sedgewick and K. Wayne, Addison-Wesley Professional, 2011, ISBN 0–321–57351–X.



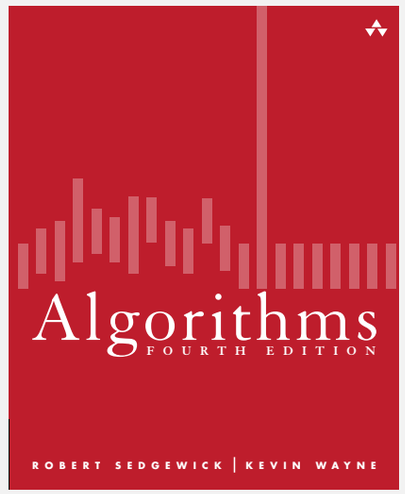
1st edition (1982)



2nd edition (1988)



3rd edition (1997)



4th edition (2011)

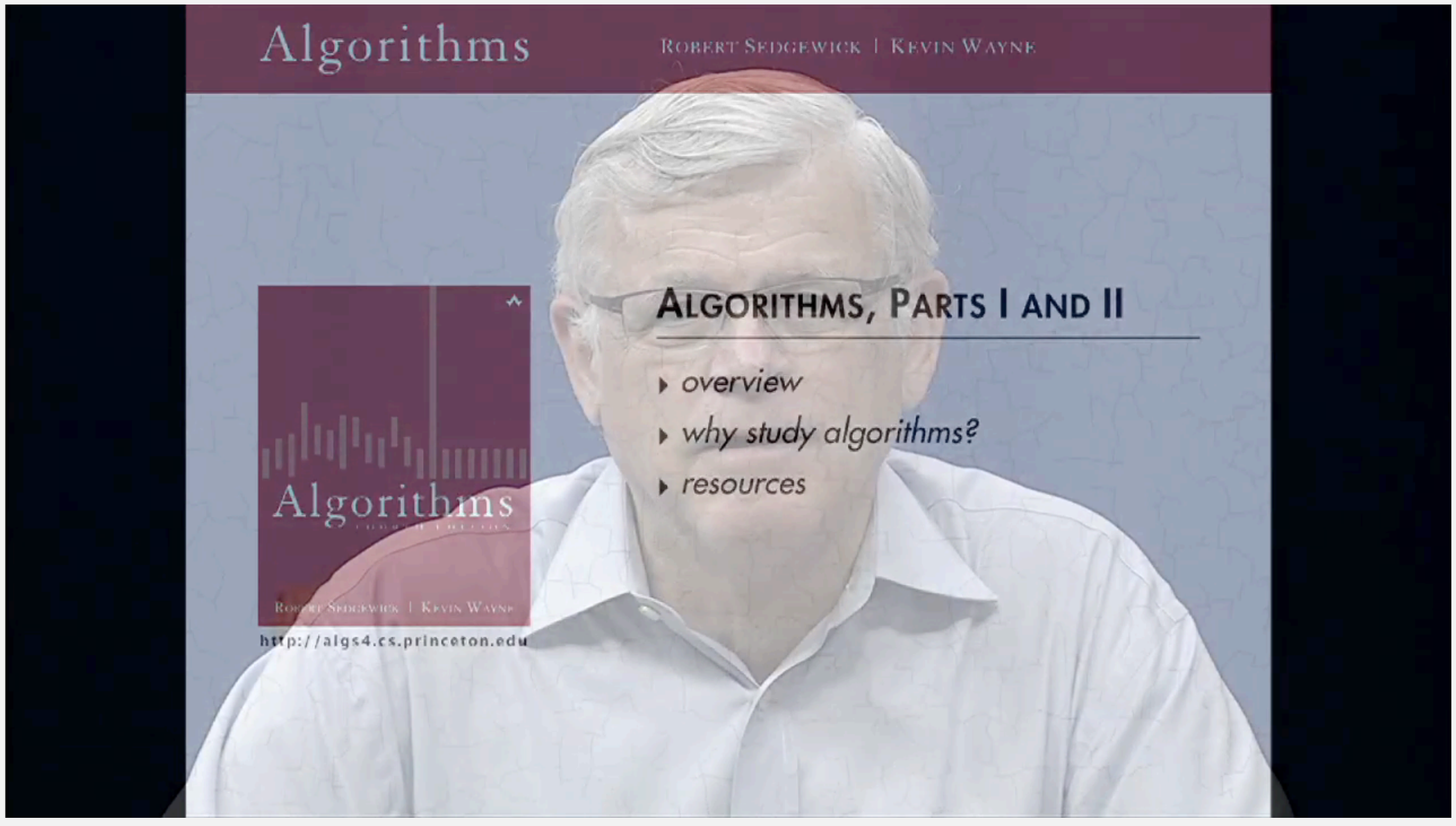
Available from various vendors and formats.

- Amazon: $70 hardcover, $58 Kindle, ...
- Labyrinth: $63 hardcover, $40 rent.
- Engineering library: on reserve.
- Safari Tech Books Online.

Resources (videos)

Lecture videos (optional).

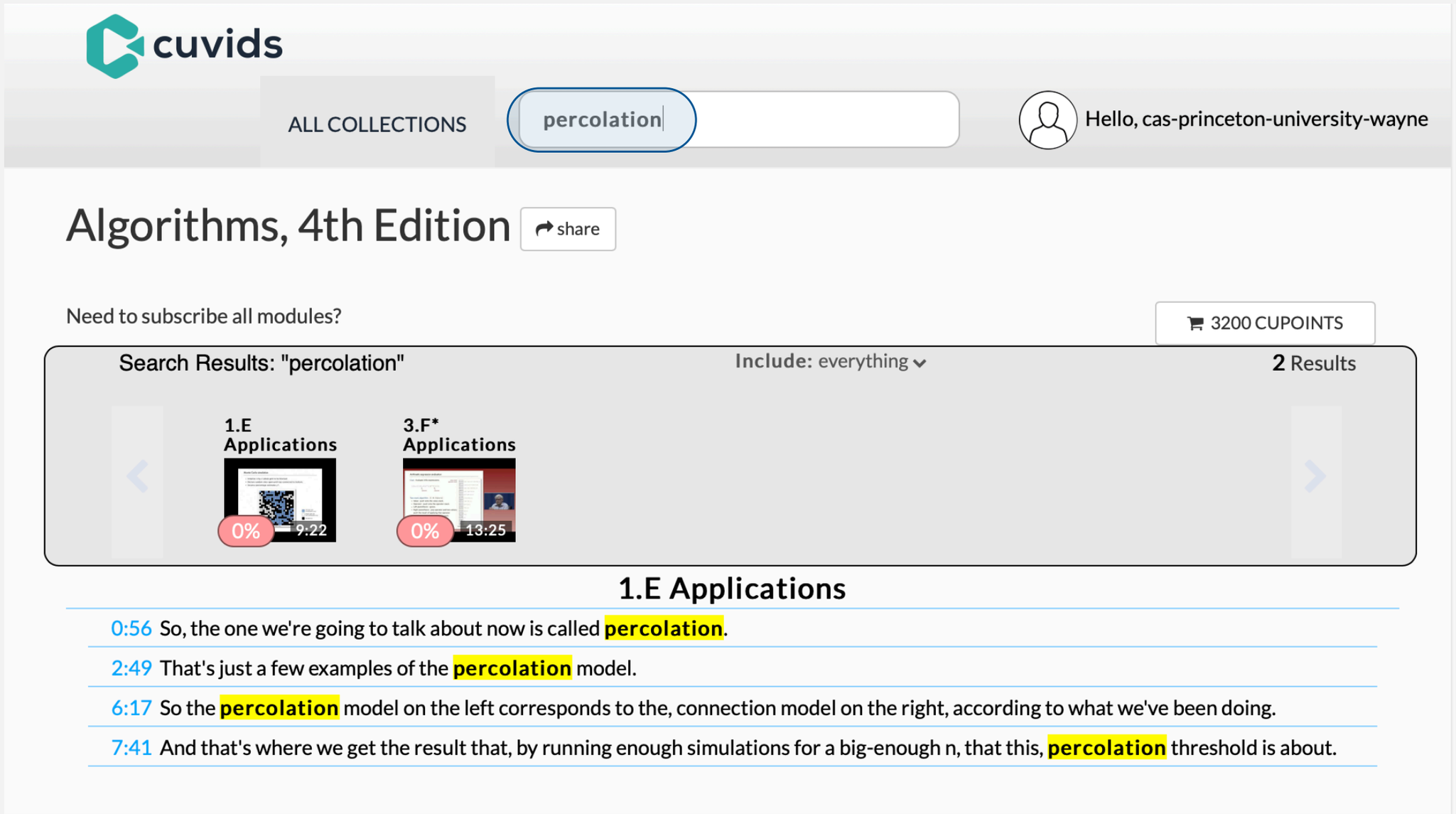
- Missed lecture.
- Review for exams.



Resources (videos)

Lecture videos (optional).

- Missed lecture.
- Review for exams.

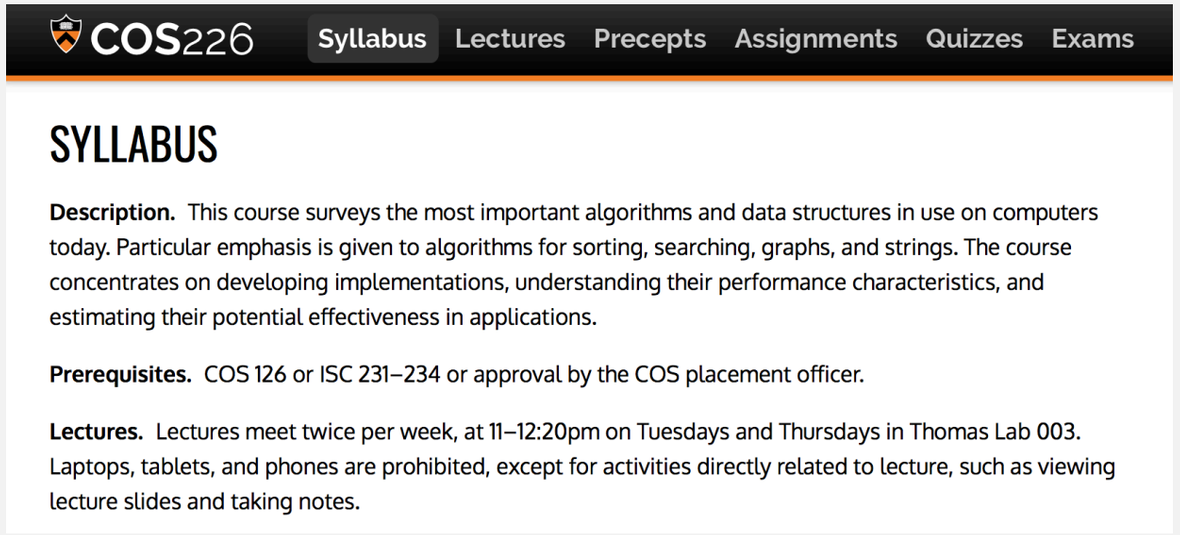


https://cuvids.io

Resources (web)

Course content.

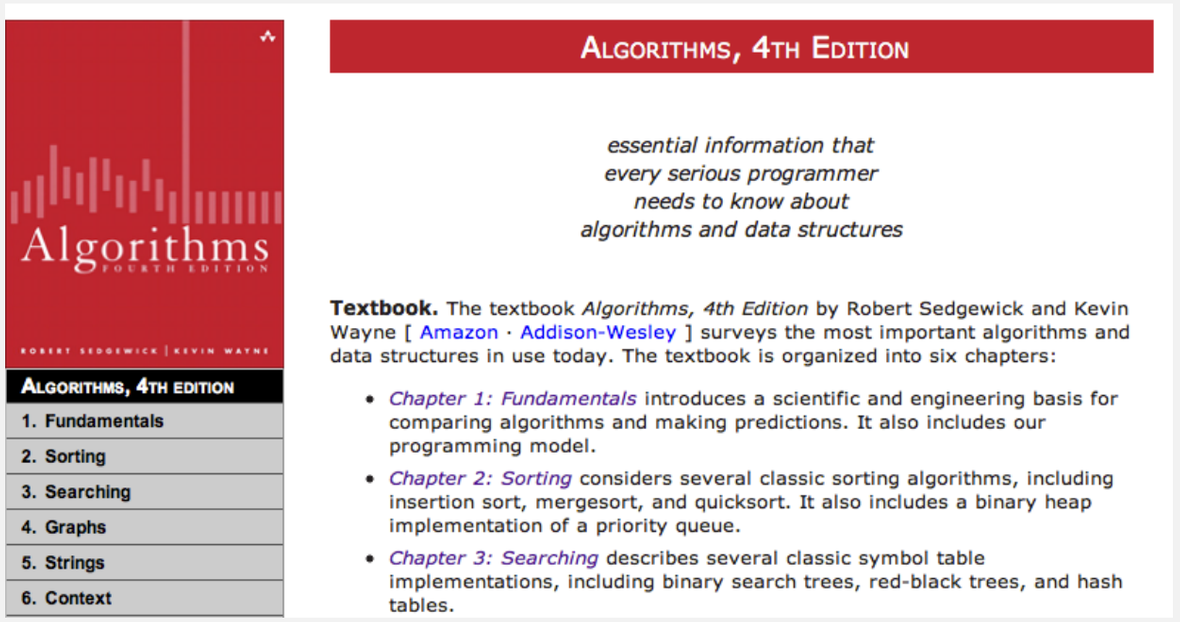
- Course info.
- Lecture slides.
- Programming assignments.
- Quizzes.
- Exam archive.



https://www.cs.princeton.edu/~cos226

Booksite.

- Brief summary of content.
- Download code from book.
- APIs and Javadoc.



https://algs4.cs.princeton.edu

Resources (people)

Online discussion forum.

- Low latency, low bandwidth.
- See Ed for guidelines.
- Use Ed; do not email course staff.

https://us.edstem.org/courses/77

Office hours.

- High bandwidth, high latency.
- See web for schedule.

https://www.cs.princeton.edu/~cos226

Computing laboratory.

- Undergrad lab TAs.
- For help with debugging.
- See web for schedule.

https://labta.cs.princeton.edu

A typical week (including this one!)

you are here!

support lecture material;
assignment prep

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4 **Lecture 1 (Union-Find)**	5	6 **Lecture 2 (Analysis)**	7 **Precept 1** **Quiz 1 and 2**	8
9	10 **Assignment 1 (Percolation)**	11	12	13	14	15

content based on
week's material

content based on
corresponding lectures

Q+A

Not registered? Register ASAP; attend any precept this week.

Change precept? Use TigerHub.

All non-conflicting precepts closed? See Colleen Kenny in CS 210.

Haven't taken COS 126? See COS placement officer.

Placed out of COS 126? Review Sections 1.1–1.2 of Algorithms 4/e.