

COS 226, SPRING 2022

ALGORITHMS  
and  
DATA STRUCTURES

KEVIN WAYNE · DAN LEYZBERG · JÉRÉMIE LUMBROSO



PRINCETON  
UNIVERSITY

# FINE PRINT



*I will be recording lectures and make them available in Canvas.*

*Because of privacy, compliance, and legal considerations,  
you may not record or redistribute recordings of this class.*

<https://itpolicy.princeton.edu/guidelines>





<https://algs4.cs.princeton.edu>

## INTRO TO COS 226

---

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



<https://algs4.cs.princeton.edu>

# INTRO TO COS 226

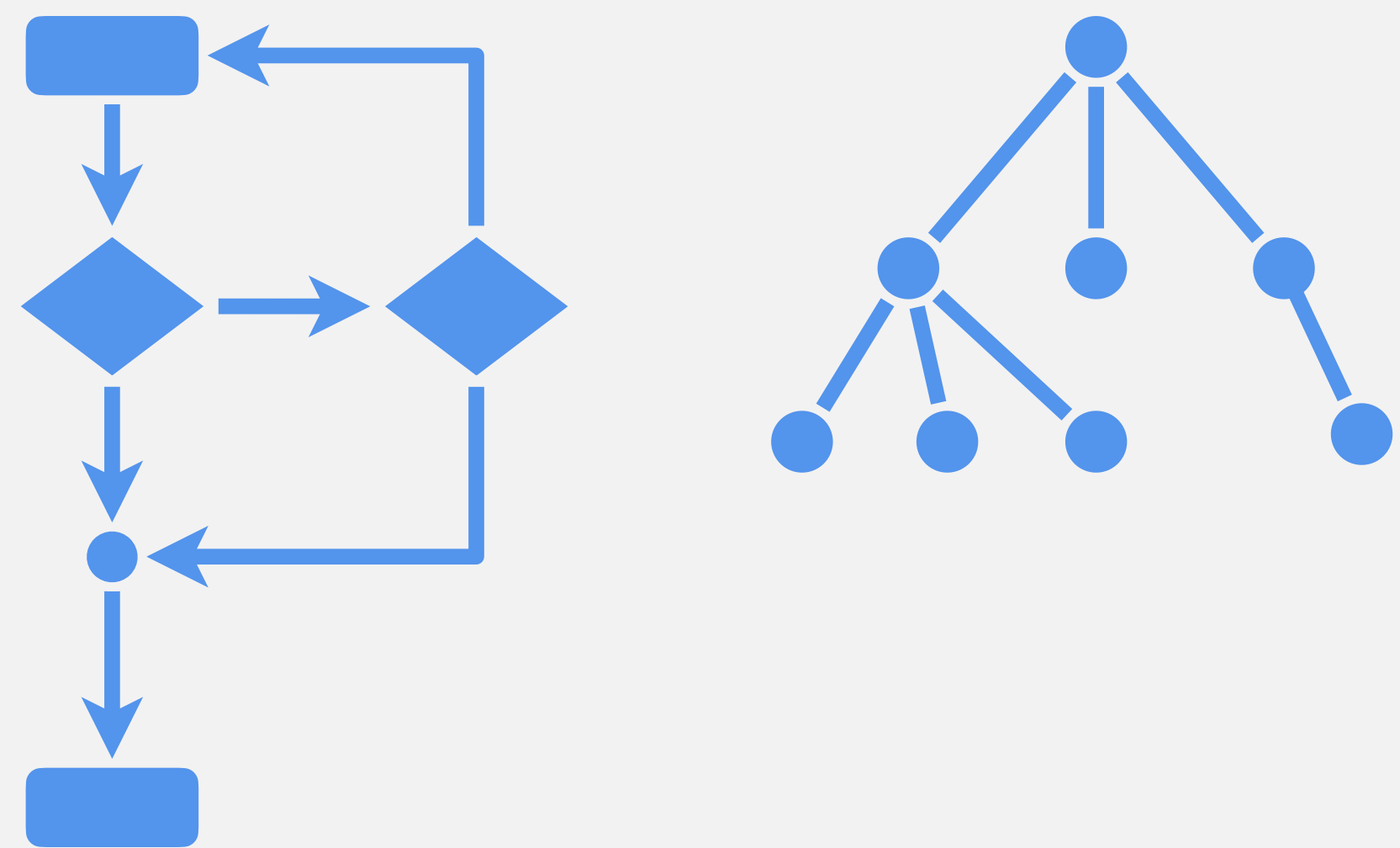
---


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

# COS 226 course overview

## What is COS 226?

- Intermediate-level survey course.
- Programming and problem solving, with applications.
- **Algorithm:** step-by-step procedure for solving a problem.
- **Data structure:** method for organizing data in a computer.

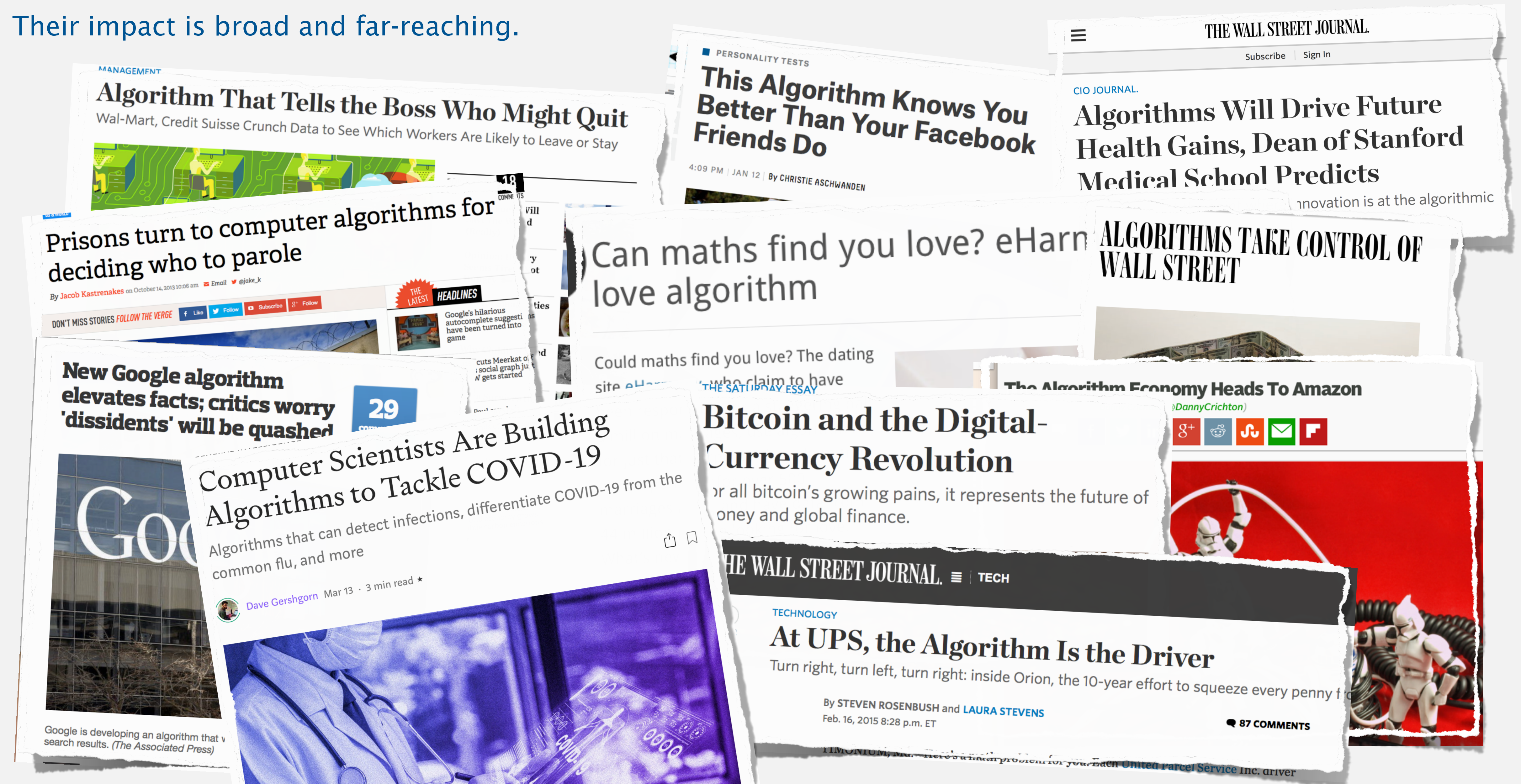


topic	algorithms and data structures 
data types	<i>stack, queue, union-find</i>
sorting	<i>quicksort, mergesort, heapsort, priority queue</i>
searching	<i>BST, red-black BST, hash table, k-d tree</i>
graphs	<i>BFS, DFS, Prim, Kruskal, Dijkstra, Bellman-Ford</i>
strings	<i>radix sorts, tries, suffix arrays, data compression</i>



# Why study algorithms and data structures?

Their impact is broad and far-reaching.





# Why study algorithms and data structures?

---

To become a proficient programmer.

```
:n==t?this.pause().cycle():this.slide
ent.find(".next, .prev").length&&e.su
clearInterval(this.interval),this.int
nction(){if(this.sliding)return;retur
|r[t]() ,s=this.interval,o=t=="next"?
h?i:this.$element.find(".item")[u](),
cators.length&&(this.$indicators.find
children()[a.getActiveIndex()]);t&&t.
t.trigger(f);if(/<\/>defaultPrevented(
t.transition(n){i.removeC
iding=!1,se on(){a.$ele
moveClass("active","acti
.carouselTransition(n){r
defaults(n),o=typ
[o]():s.in .cycle()})
t,e.fn.carousel=t=function()
slide-to]",fun var n=e(this),
({},i.data( carousel(s
}))}(wind e){use
ts,n). is.$par
```

*“ I will, in fact, claim that the difference between a bad programmer and a good one is whether [they] consider [their] code or [their] 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.

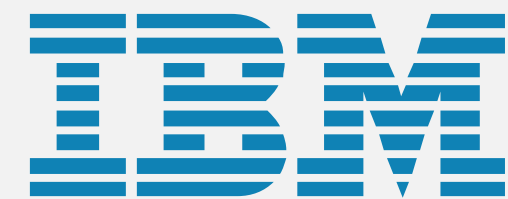


*“For me, **great algorithms** are the poetry of computation. Just like verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing.” — Francis Sullivan*

# Why study algorithms and data structures?

---

For fun and profit.



NVIDIA®



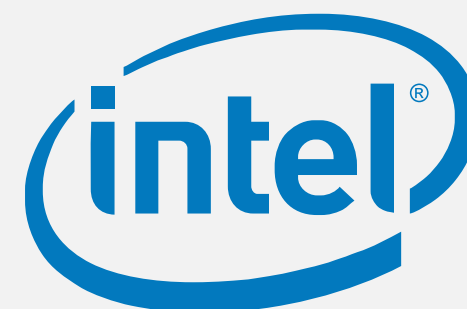
Adobe



UBER



ORACLE®



Microsoft®



# Why study algorithms and data structures?

---

- Their impact is broad and far-reaching.
- To become a proficient programmer.
- For intellectual stimulation.
- For fun and profit.

Why study anything else?






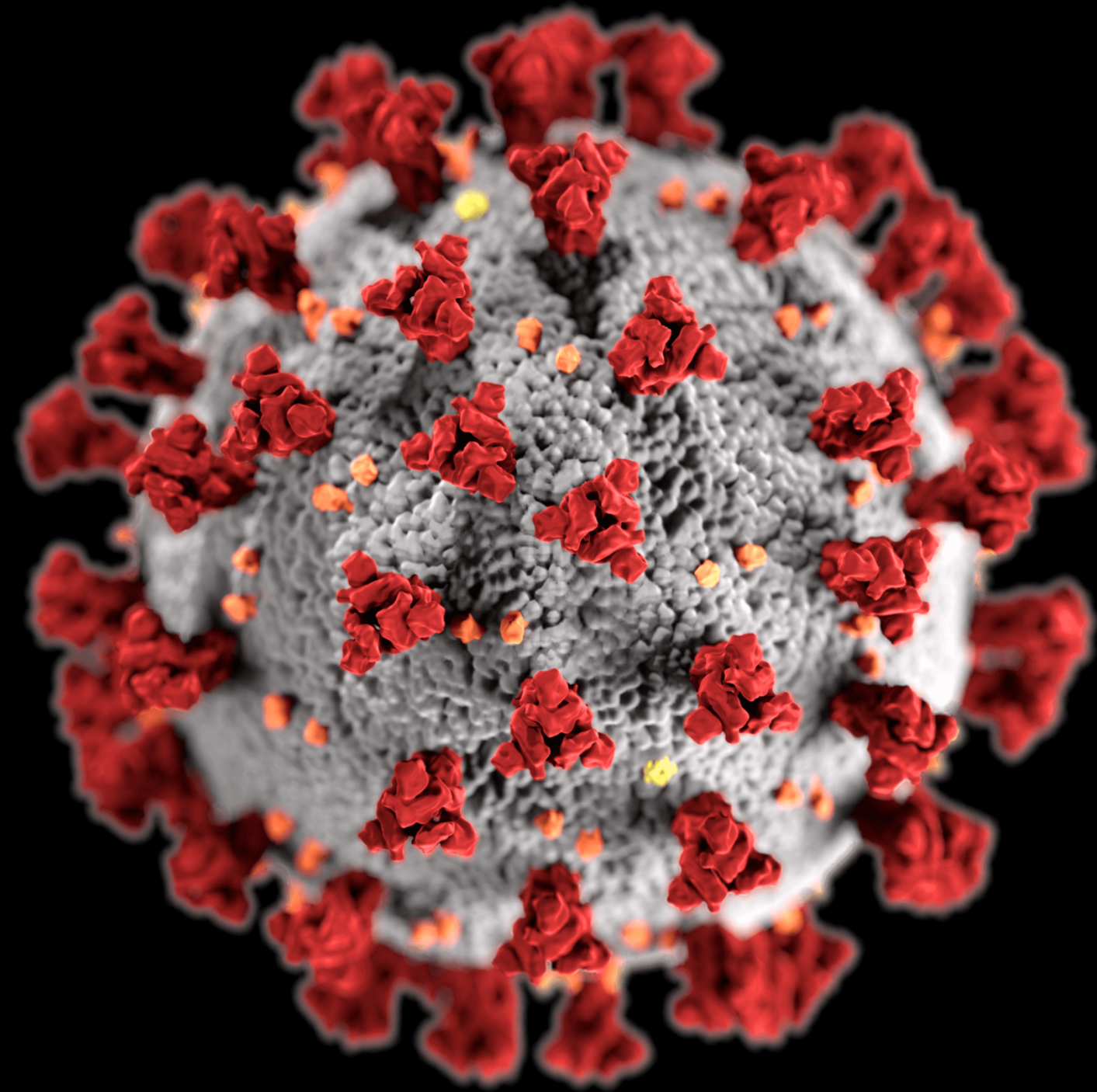


<https://algs4.cs.princeton.edu>

# INTRO TO COS 226

---

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



*course format subject to change*



# Lectures

Live lectures. Introduce new material.

What	When	Where	Who	Office Hours
L01	TTh 11-12:20pm	Friend 101	Kevin Wayne	<i>see web</i>

Questions. Don't be bashful.

Electronic devices. Permitted *only* to support lecture.

viewing slides, taking notes, iClickers, ...



iClicker (required). To earn participation credit:

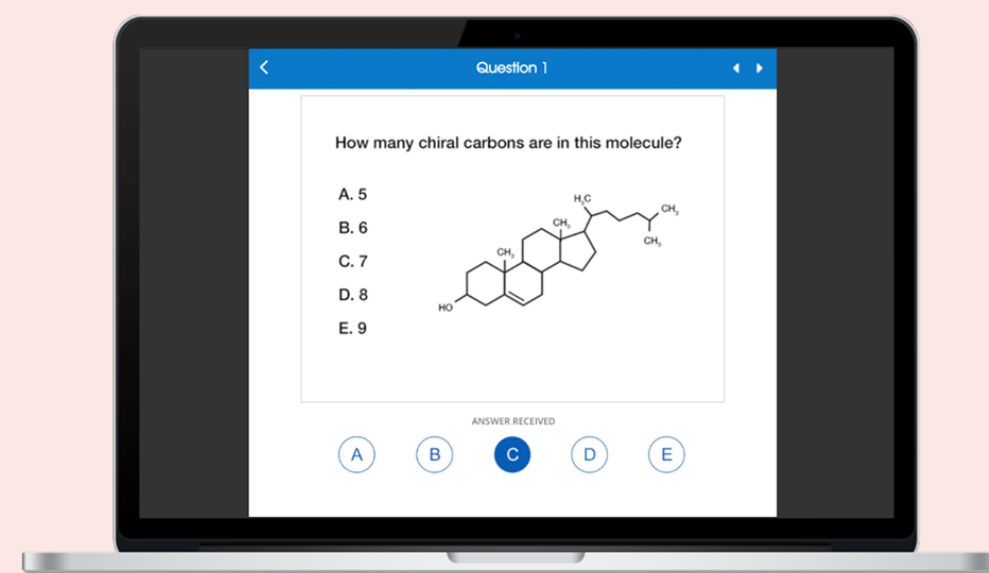
- Answer multiple choice questions during lecture.
- Register either hardware remote or web/mobile app.
- Use only one device per lecture.

free for Princeton students

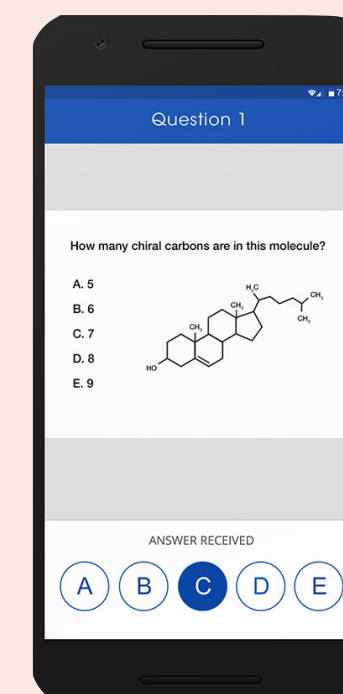
Which iClicker device are you using?



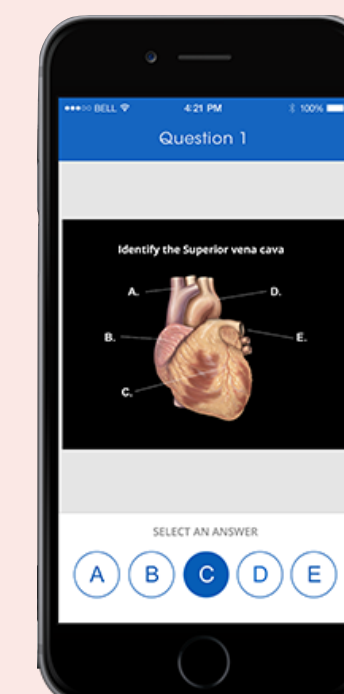
A. Hardware



B. Web



C. iPhone



D. Android



# Precepts

Active learning. Problem-solving, discussion, assignment prep, ...



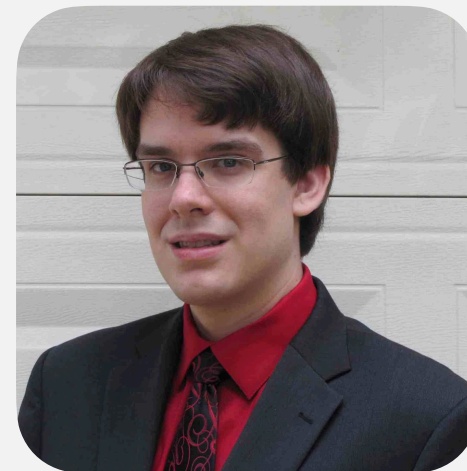
Dan Leyzberg



Gabriel Contreras



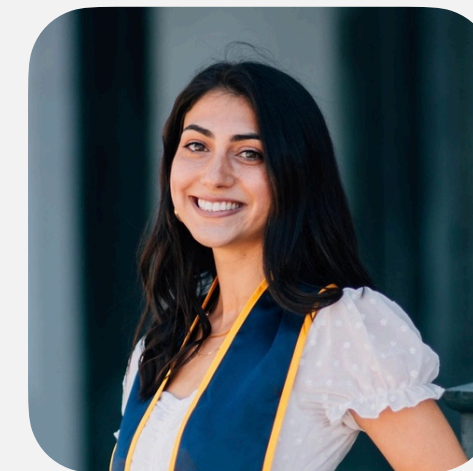
Weicong Dong



Ryan Torok



Yingxi Lin











Morgan Nanez




Max Tchouambe

ed

COS 226 – Lessons



[Lessons](#) [Slides](#) [Prev](#) [Next](#)

 Union-Find

Edit Slide ...

Ah, our good friend, the union-find. Who ever thought that such a simple data type could lead us to learn about so many clever algorithmic design tricks? On this page, we'll take a closer look at some of those tricks and how you might apply them to other problems.

**Question 1**

For our first approach, we did what you might do if you were trying to keep track of merged sets by hand: assign each set some kind of name, and, when you merge two sets, replace all instances of one name with the other.

Speaking of which, how would you describe what quick-find stores in the `leader[]` array? What are those numbers, exactly?

200

# Precepts

---

What	When	Where	Who
P01	Th 3–4:20pm	TBA	TBA
P02	Th 7:30–8:50pm	TBA	TBA
P03	F 11–12:20pm	TBA	TBA
P04	F 11–12:20pm	TBA	TBA
P05	F 1:30–2:50pm	TBA	TBA
P06	F 1:30–2:50pm	TBA	TBA
P07	F 3–4:20pm	TBA	TBA
P08	F 3–4:20pm	TBA	TBA
P09	F 11–12:20pm	TBA	TBA

**CANCELLED**



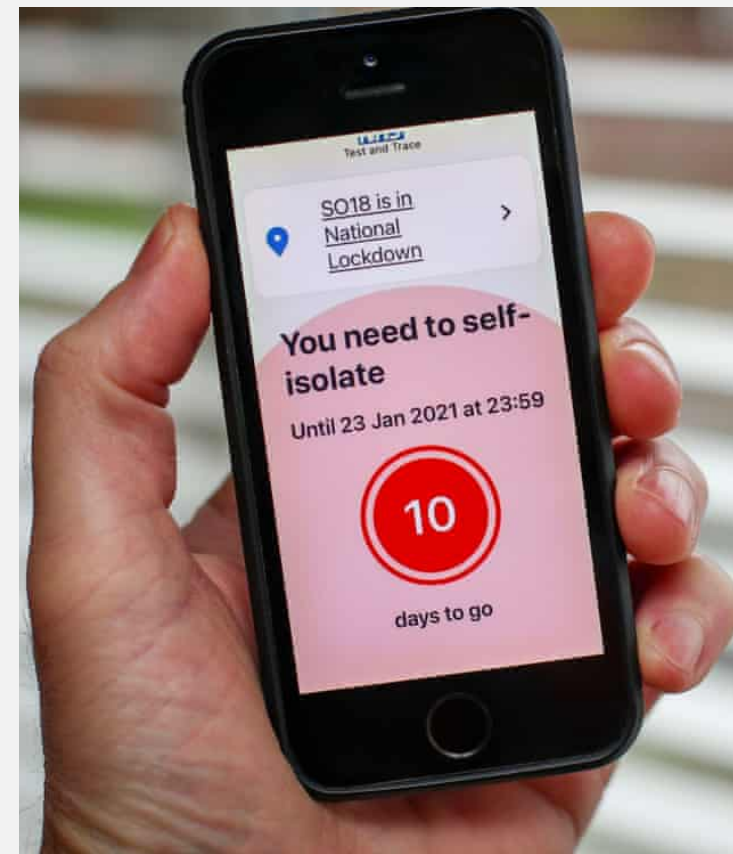
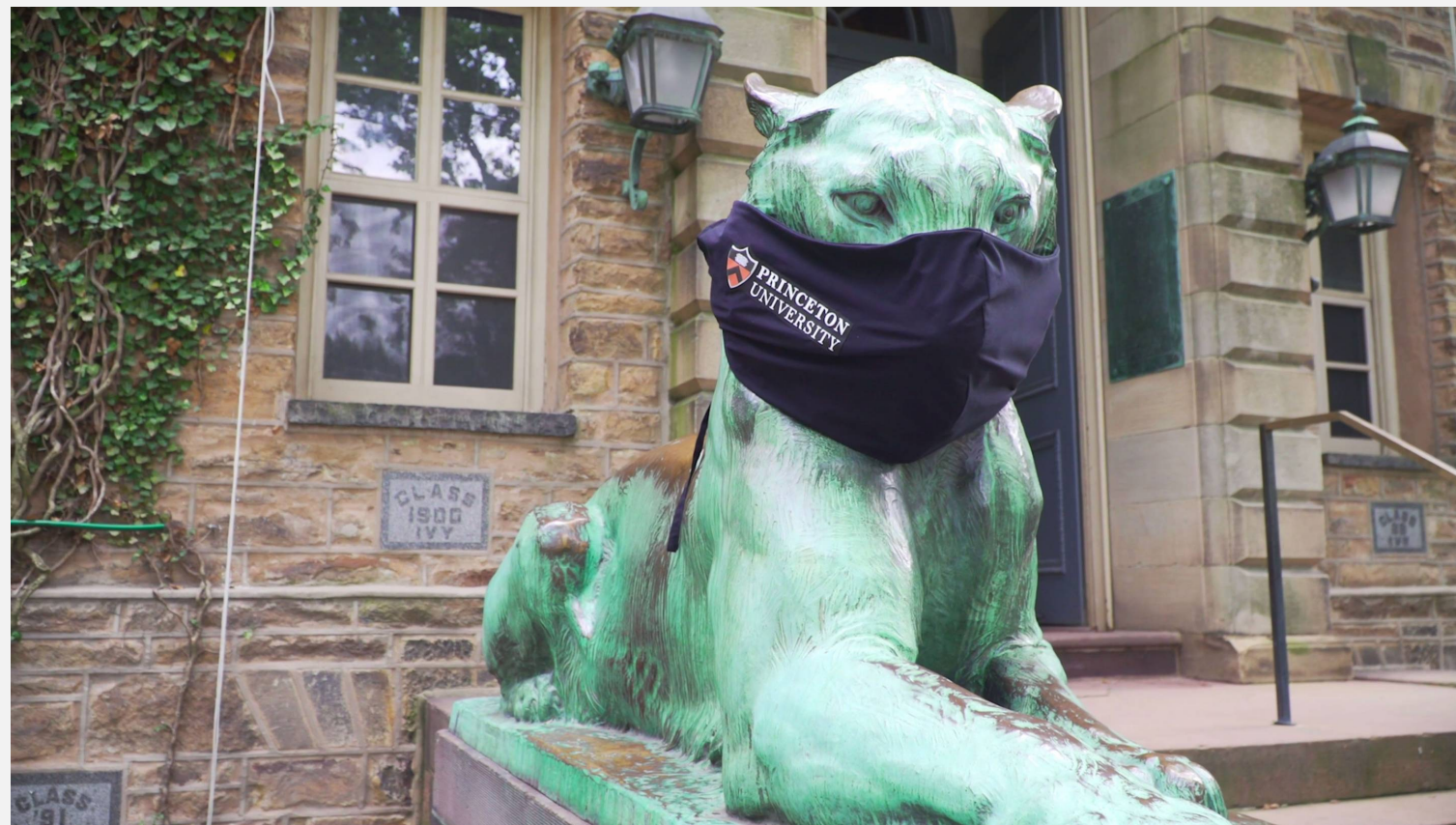
# Covid-19 policies

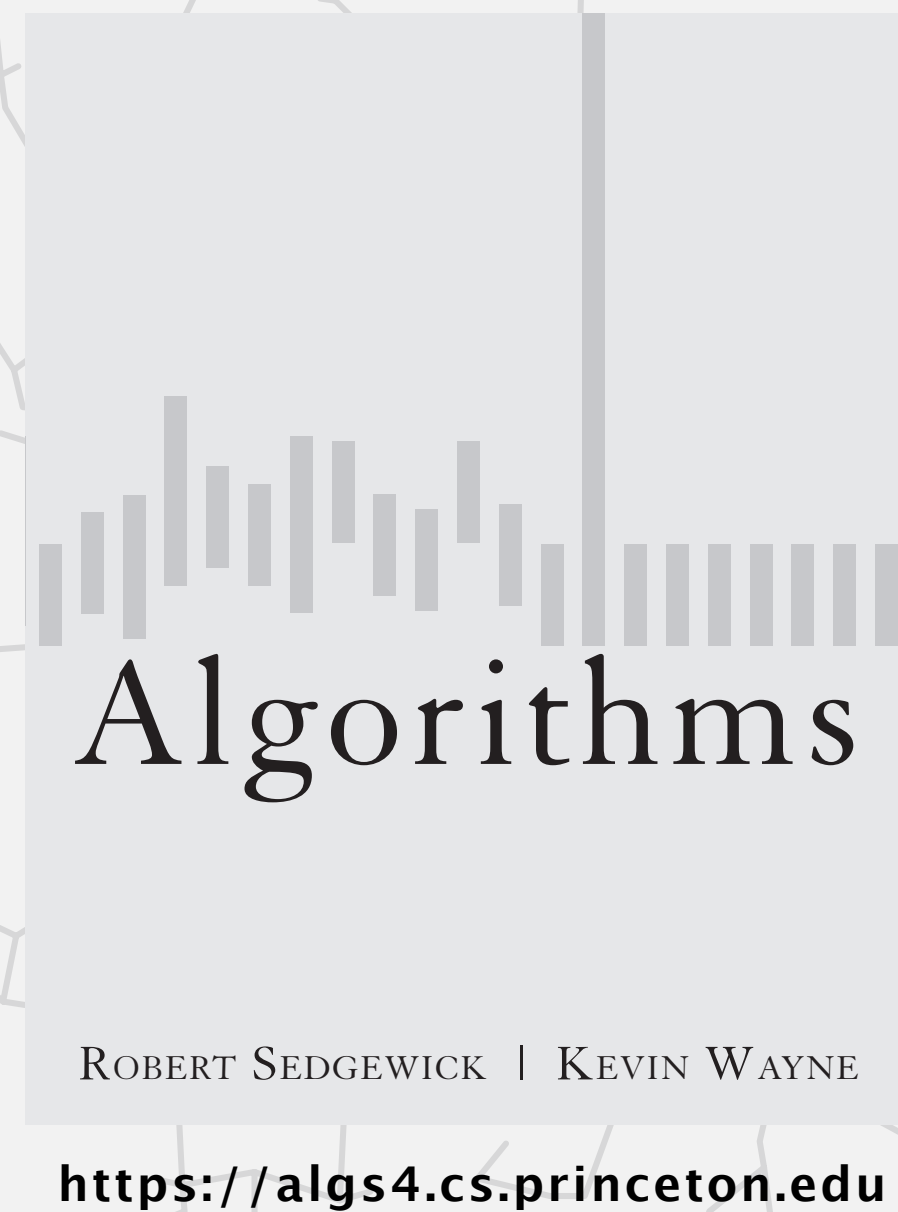
---

Face coverings. Abide by university rules.

Symptomatic. Stay home and call UHS.

Self-isolation for students. Lecture recordings in Canvas.





# INTRO TO COS 226

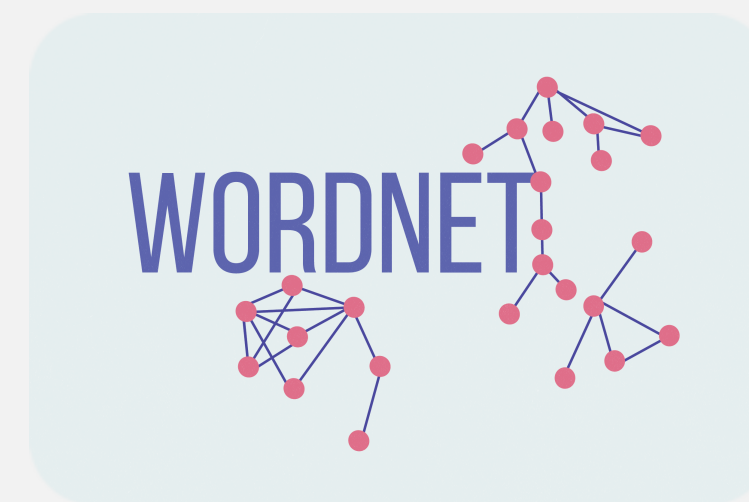
---

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

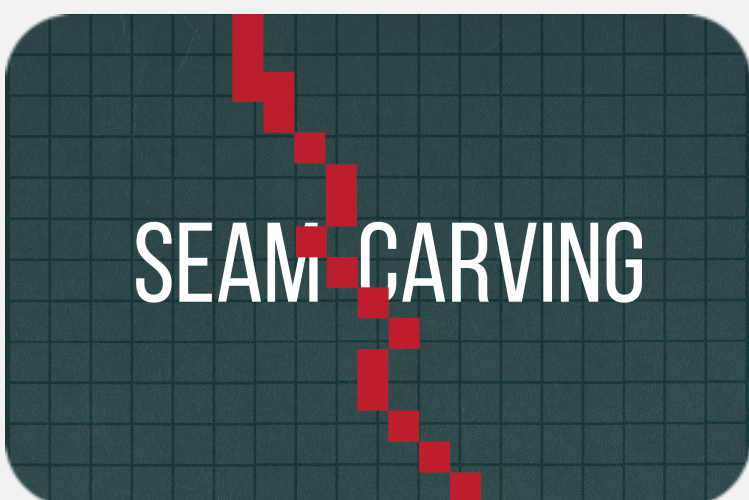


# Programming assignments

Implement an efficient **algorithm** or **data structure**:



Solve an interesting **application** using a “textbook” algorithm:



**Pair programming** encouraged on designated assignments.

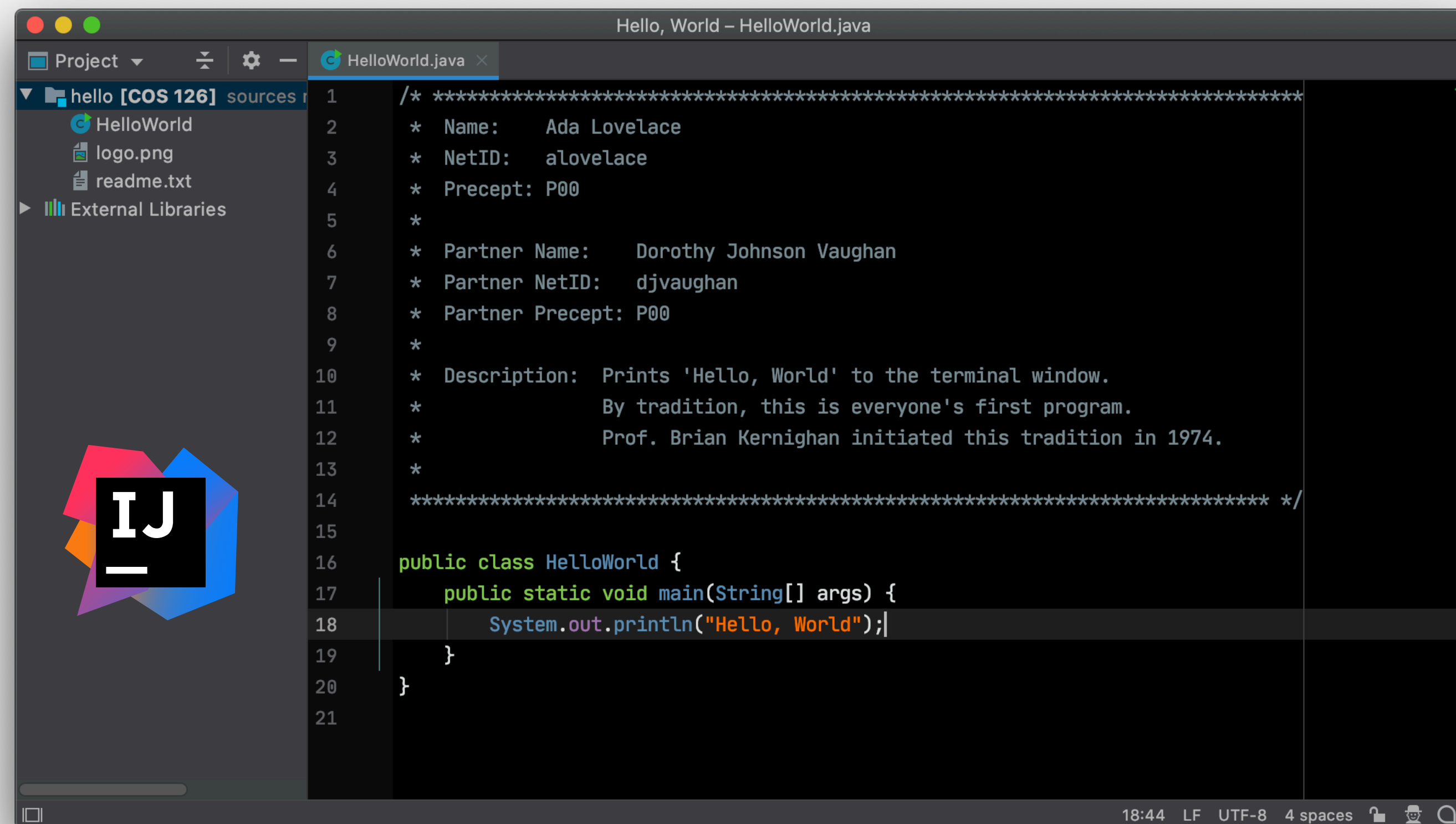




# Programming environment


**Recommended IDE.** Custom IntelliJ 2021.1 environment.  upgrade to Fall 2021 version

- Embedded Bash terminal.
- Autoformat, autoimport, autocomplete, ....
- Continuous code inspection; integrated Checkstyle and SpotBugs.
- ...



## Quizzera platform.

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



### Quizzera

wayne@cs.princeton.edu [Logout](#)

[Dashboard](#) / [COS 226, Spring 2020](#) / [Union Find](#) / [Quick Find](#) / **Attempt**

#### Quick Find

Seed: **235325** (Provider: [princeton.cos226.QuickFindExercise](#))

Attempts Remaining: **2** Quiz Ends in **4 days**.

[New Attempt](#)

#### Question

Give the `id[]` array that results from the following sequence of 6 union operations on a set of 10 items using the quick-find algorithm.

8-4 1-0 1-4 4-2 7-8 9-1

Recall: our quick-find convention for the union operation `p-q` is to change `id[p]` (and perhaps some other entries) but not `id[q]`.

#### Answer

Your answer should be a sequence of 10 integers (between 0 and 9), separated by whitespace.

[Submit](#)

## Written exams.

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

### Q4 Analysis of algorithms

8 Points



Consider a *zig-zag array* that contains the integers 1 through  $n/2$  in ascending order, interleaved with  $n/2$  copies of the integer 0, where  $n$  is an even integer. For example, here is the array when  $n = 16$ :

0 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8

#### Q4.1 Selection sort

2 Points

How many *compares* does *selection sort* make to sort a zig-zag array as a function of  $n$ ?

- ☐  $\sim \frac{1}{16}n^2$
- ☐  $\sim \frac{1}{8}n^2$
- ☐  $\sim \frac{1}{4}n^2$
- ☒  $\sim \frac{1}{2}n^2$
- ☐  $\sim n^2$

# Grading **A+**

---

## Programming assignments. **45%**

- Due at 11:59pm on Mondays via TigerFile.
- Collaboration/lateness policies: see web.

## Quizzes. **10%**

- Due at 11:59pm on Fridays via Quizzera.
- Collaboration/lateness policies: see web.

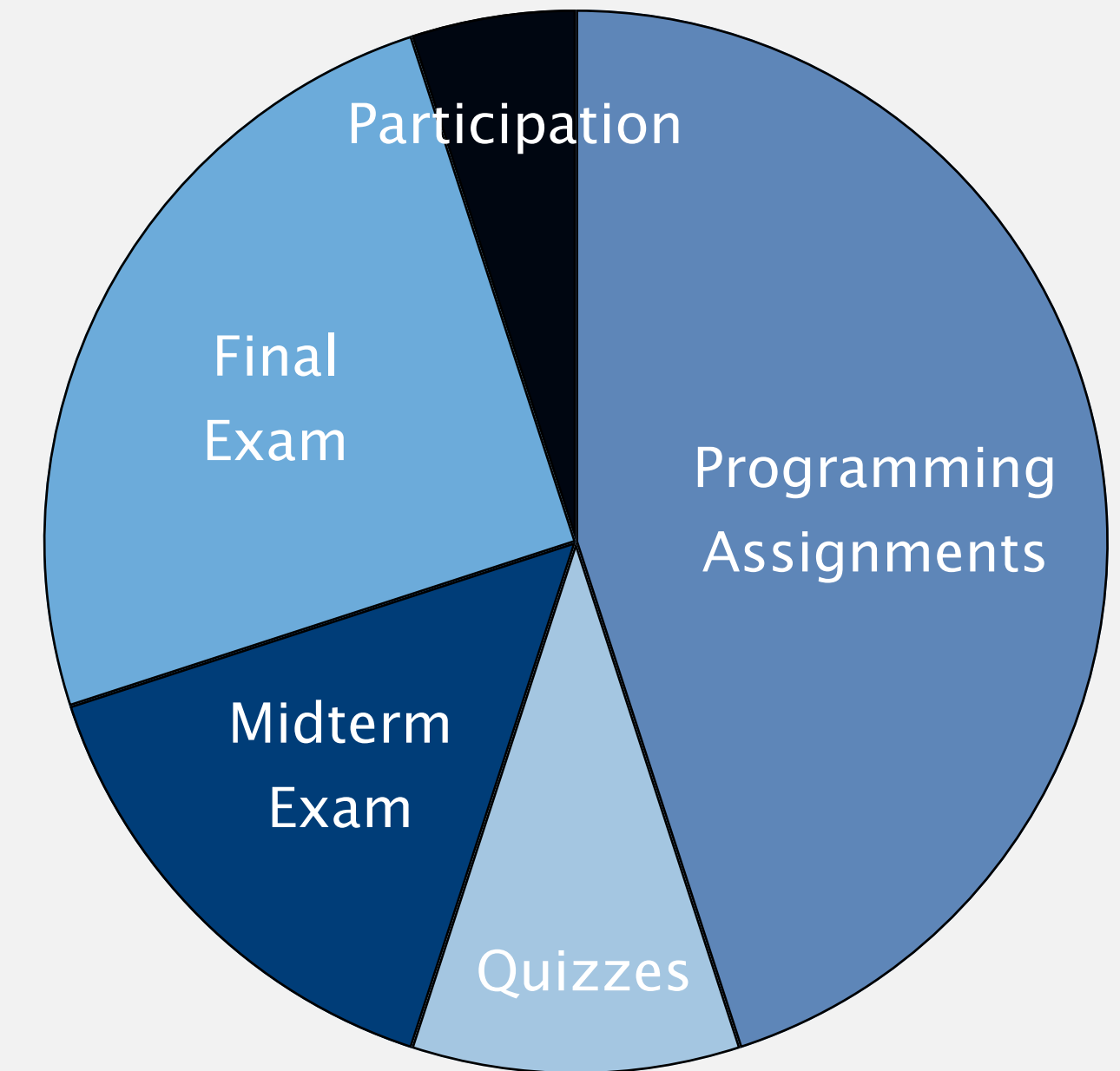
## Exams. **15% + 25%**

- 80-minute in-class (or remote) midterm on Thursday, March 3.
- 3-hour in-class (or remote) final, as scheduled by Registrar.

## Active participation. **5%**

- Answer questions in online discussion forum.
- iClicker participation in lecture.

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





<https://algs4.cs.princeton.edu>

# INTRO TO COS 226

---

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



# Resources (textbook)

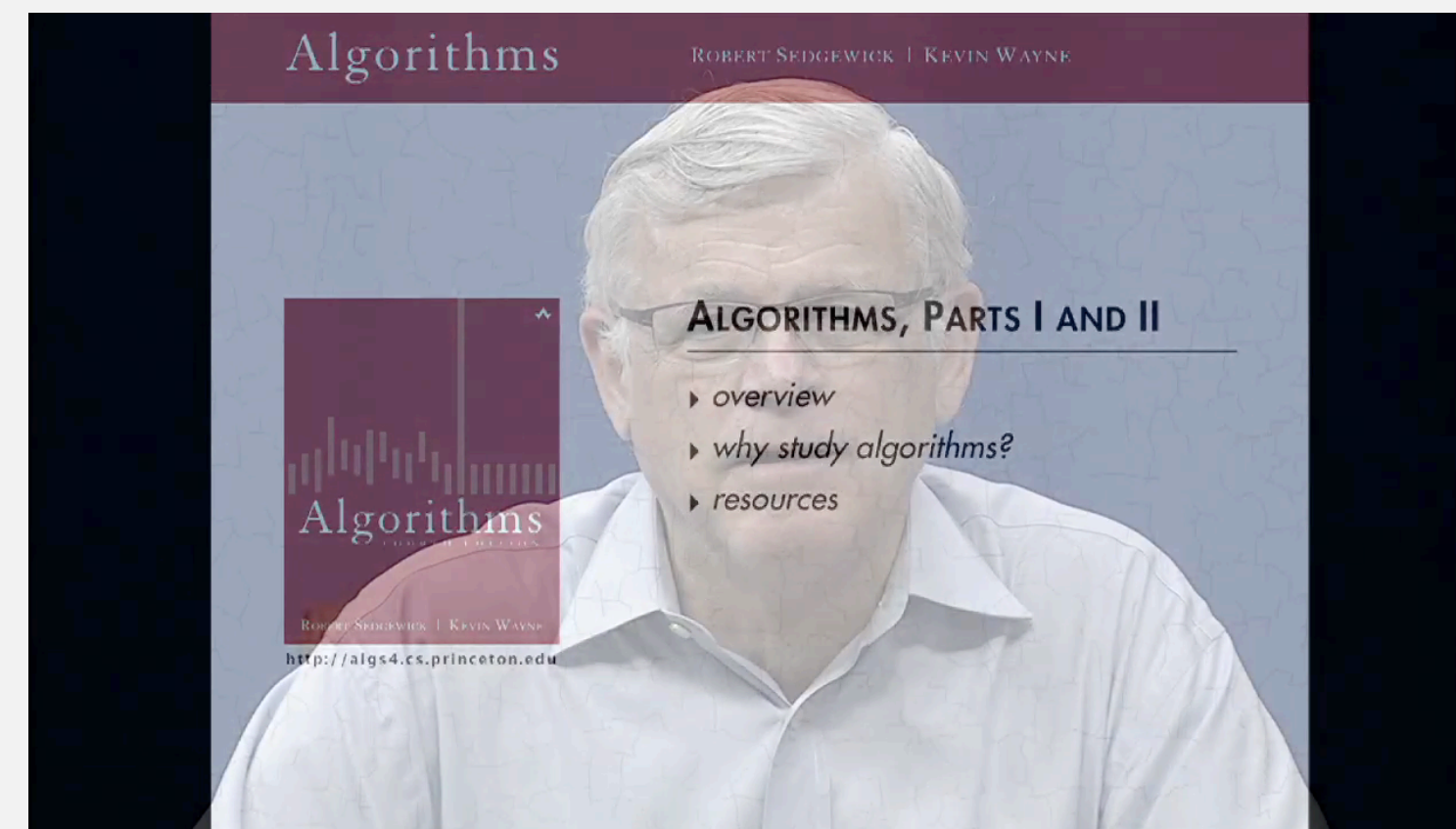


**Readings (required).** *Algorithms 4<sup>th</sup> edition* by R. Sedgewick and K. Wayne, Addison-Wesley Professional, 2011, ISBN 0–321–57351–X.

**Studio-produced videos (optional).** By R. Sedgewick and K. Wayne.




4<sup>th</sup> edition (2011)



<https://www.cubits.ai/collections/42>

## Course content.

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

 **COS226** [Syllabus](#) [Lectures](#) [Precepts](#) [Assignments](#) [Quizzes](#) [Exams](#)

### SYLLABUS


**Description.** This course surveys the most important algorithms and data structures in use on computers today. Particular emphasis is given to algorithms for sorting, searching, graphs, and strings. The course concentrates on developing implementations, understanding their performance characteristics, and estimating their potential effectiveness in applications.

**Prerequisites.** COS 126 or ISC 231–234 or approval by the COS placement officer.

<https://www.princeton.edu/~cos226>

## Booksite.

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



ALGORITHMS, 4TH EDITION
1. Fundamentals
2. Sorting
3. Searching
4. Graphs
5. Strings
6. Context

### ALGORITHMS, 4TH EDITION

*essential information that  
every serious programmer  
needs to know about  
algorithms and data structures*

**Textbook.** The textbook *Algorithms, 4th Edition* by Robert Sedgwick and Kevin Wayne [ [Amazon](#) · [Addison-Wesley](#) ] surveys the most important algorithms and data structures in use today. The textbook is organized into six chapters:

- *Chapter 1: Fundamentals* introduces a scientific and engineering basis for comparing algorithms and making predictions. It also includes our programming model.
- *Chapter 2: Sorting* considers several classic sorting algorithms, including insertion sort, mergesort, and quicksort. It also includes a binary heap implementation of a priority queue.
- *Chapter 3: Searching* describes several classic symbol table implementations, including binary search trees, red-black trees, and hash tables.

<https://algs4.cs.princeton.edu>



## Online discussion forum.

- Low latency, low bandwidth.
- Mark post private when necessary.
- See Ed FAQ for guidelines.



<https://us.edstem.org/courses/18108>

## Office hours.

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













<https://www.princeton.edu/~cos226>

## “Computing laboratory.”

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

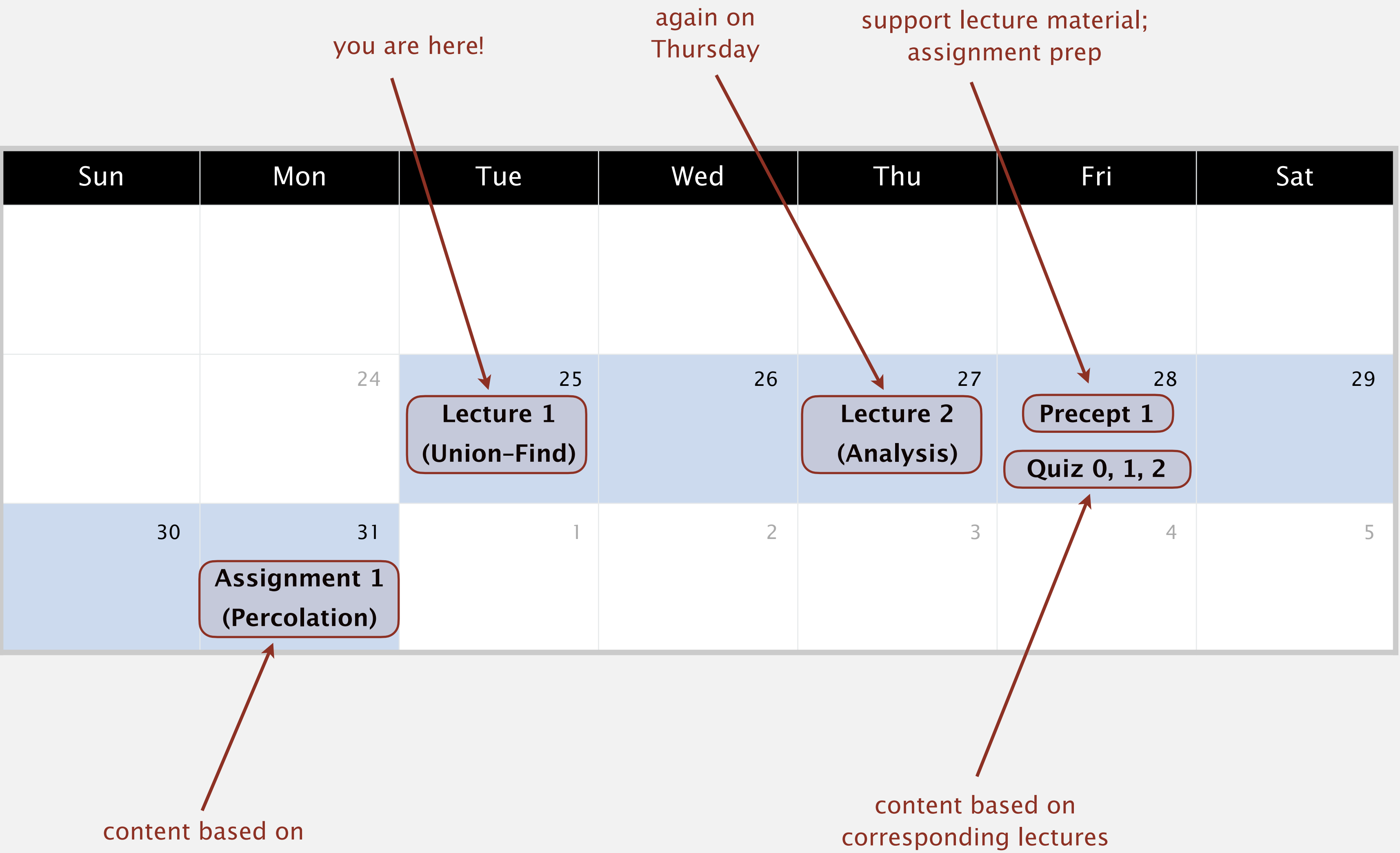


<https://labta.cs.princeton.edu>

Platform	What
 <b>Ed</b>	<i>discussion forum, precept lessons</i>
 <b>IntelliJ</b>	<i>Java IDE</i>
 <b>Zoom</b>	<i>some office hours</i>
 <b>Quizzera</b>	<i>quizzes</i>
 <b>TigerFile</b>	<i>assignment submissions</i>
 <b>codePost</b>	<i>assignment feedback</i>
 <b>Gradescope</b>	<i>remote exams</i>
 <b>Canvas</b>	<i>grades, lecture recordings</i>
 <b>iClicker</b>	<i>in-class polls</i>
 <b>CUbits</b>	<i>studio-produced videos</i>

← also use for communication with course staff

# A typical week (including this one!)



## Administrative Q+A

---

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

**Change precept?** Use TigerHub.

**All non-conflicting precepts closed?** Contact Colleen Kenny.

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

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

**Additional administrative questions.** Ask now, after class, or any time in Ed Discussion.

