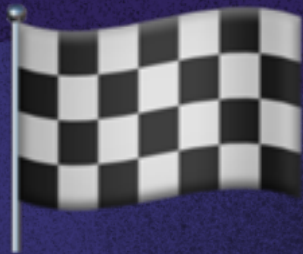


LEC 24

**CSE 373**

# Course Wrap-Up



BEFORE WE START

*Use the Zoom chat:*

**What are you looking forward to  
for the rest of your summer?**

*and, more importantly:*

**Now that we're done with the  
quarter, what's your favorite data  
structure?**

---

**Instructor** Aaron Johnston

**TAs**

Timothy Akintilo  
Brian Chan  
Joyce Elauria  
Eric Fan  
Farrell Fileas

Melissa Hovik  
Leona Kazi  
Keanu Vestil  
Siddharth Vaidyanathan  
Howard Xiao

# Announcements – Almost There!

- EX4 late cutoff tomorrow night
- P4 due tonight, late cutoff Saturday night
  - If using late days, **please** plan ahead with the exam due the same night!
- Extra credit also due Saturday night

WED	THU	FRI	SAT
<b>Course Wrap-Up</b> <i>1:10pm</i>	EX4 Late Cutoff <i>11:59pm</i>	Exam II Released <i>12:01 am</i>	<b>Exam II Due</b> <i>11:59pm</i>
<b>P4 Due</b> <i>11:59pm</i>		<b>Exam II Extra OH</b> <i>1:10pm</i>	P4 Late Cutoff, Extra Credit Due <i>11:59pm</i>

# Announcements – But First, an Exam

- Exam II
  - Released **Friday morning (8/21) at 12:01am PDT**
  - Due **Saturday evening (8/22) at 11:59pm PDT**
    - No late submission accepted – you cannot use late days on the exam!
  - Review Materials available on the course website
    - 20su-Specific Practice Problem Set + Solutions
    - Section 9 Review Worksheet
    - 19au Final Exam
    - As always, post-lecture reviews, previous section handouts, learning objectives
  - We'll publish EX3 grades and EX4 solutions before the exam
  - Focus is on post-Exam I content, although you may be asked to use Exam I skills like algorithmic analysis or early ADTs
  - Like Exam I, mostly conceptual. Unlike Exam I, you'll be required to write some code.

# Review Grading Breakdown

- Your grade will consist of the following weighted categories:
- Instead of curving the class as usual, we'll use a bucket system:
  - These are *minimum* GPA guarantees – may adjust upward

Category	Weight
Programming Projects	45%
Individual Exercises	25%
Exam I	15%
Exam II	15%

Percentage	GPA
95%	4.0
90%	3.5
80%	3.0
60%	2.0
50%	0.7



# Course Evaluations

- ***Please*** fill these out! They have an *enormous* impact on the course, and a 90% response rate makes them far more useful than 60%.
  - Have to guess what sampling bias is for “missing 40%”
- As a student, you underestimate how much instructors take evals into account – trust me, I’ve been a student here too 😊
  - Every 20su course logistics decision was the result of weeks spent looking at last quarter’s evals
- Looks like we’ll be doing this remote thing for a while, and your insight is the most valuable part of this experiment
- You all have had *amazing* response rates so far. Thanks and keep up the great work!

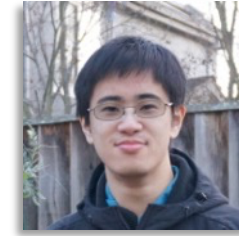
# Course Evaluations

- Evals for lecture and section are open until **Friday**
- Lecture: <https://uw.iasystem.org/survey/228731>
- Sections:
  - AA (12:00, Leona & Keanu): <https://uw.iasystem.org/survey/228950>
  - AB (1:10, Keanu & Siddharth): <https://uw.iasystem.org/survey/228937>
  - AC (1:10, Farrell & Melissa): <https://uw.iasystem.org/survey/228940>
  - AD (2:20, Joyce): <https://uw.iasystem.org/survey/228943>
  - AE (10:50, Leona & Eric): <https://uw.iasystem.org/survey/228941>
  - AF (12:00, Farrell): <https://uw.iasystem.org/survey/228944>
  - AG (1:10, Eric): <https://uw.iasystem.org/survey/228948>

*We'll end lecture a little early today so you can fill out evals!*

# The Stars of the Show

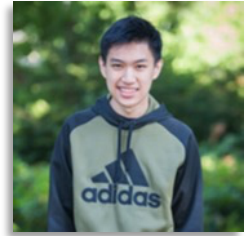
- Let's get a huge round of **zoom** applause for your TAs!
  - They do so much behind the scenes to keep 373 running smoothly... and that's on top of office hours, section, Piazza, etc.
  - None of this would be possible without them!
- UW CSE has an award for excellent TAs! Please consider nominating here:  
<https://www.cs.washington.edu/students/ta/bandes>



Brian Chan



Eric Fan



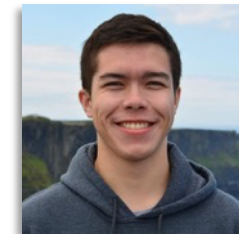
Farrell Fileas



Howard Xiao



Joyce Elauria



Keanu Vestil



Leona Kazi



Melissa Hovik



Siddharth Vaidyanathan



Timothy Akintilo

# Time For a Victory Lap

- We made it to the finish line!
- One more lap around the track by the exhausted victors (that's us 😊)
- Why?
  - It's easy to lose sight of just how much you've accomplished in a quarter!
  - Review the themes of the course that you can carry forward

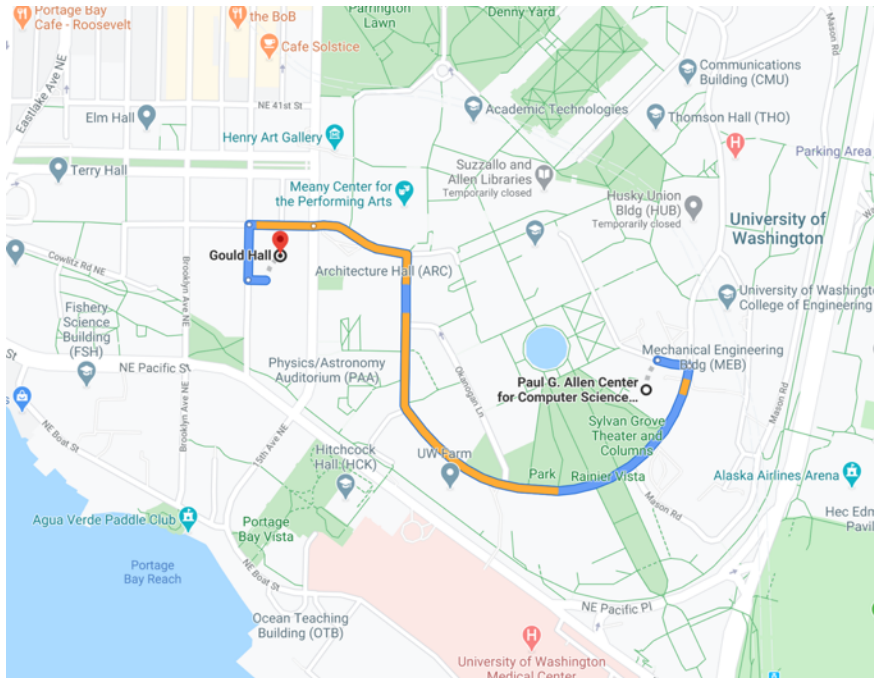




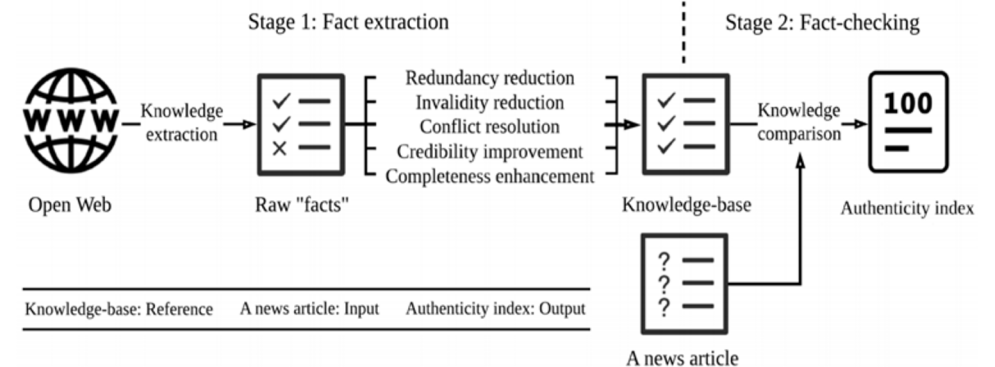
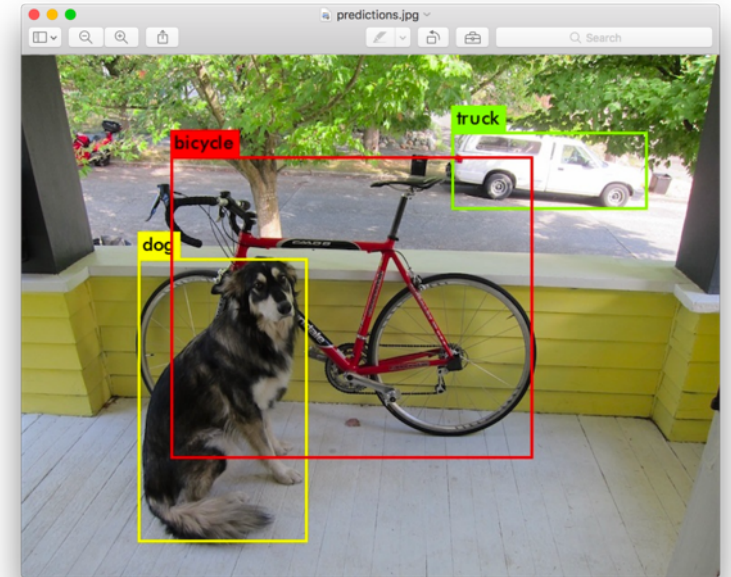
## LEC 01

# Why 373?

1. Build a **strong foundation** of data structures and algorithms that will let you tackle the biggest problems in computing



373 Data Structures  
& Algorithms



## LEC 01

# Why 373?

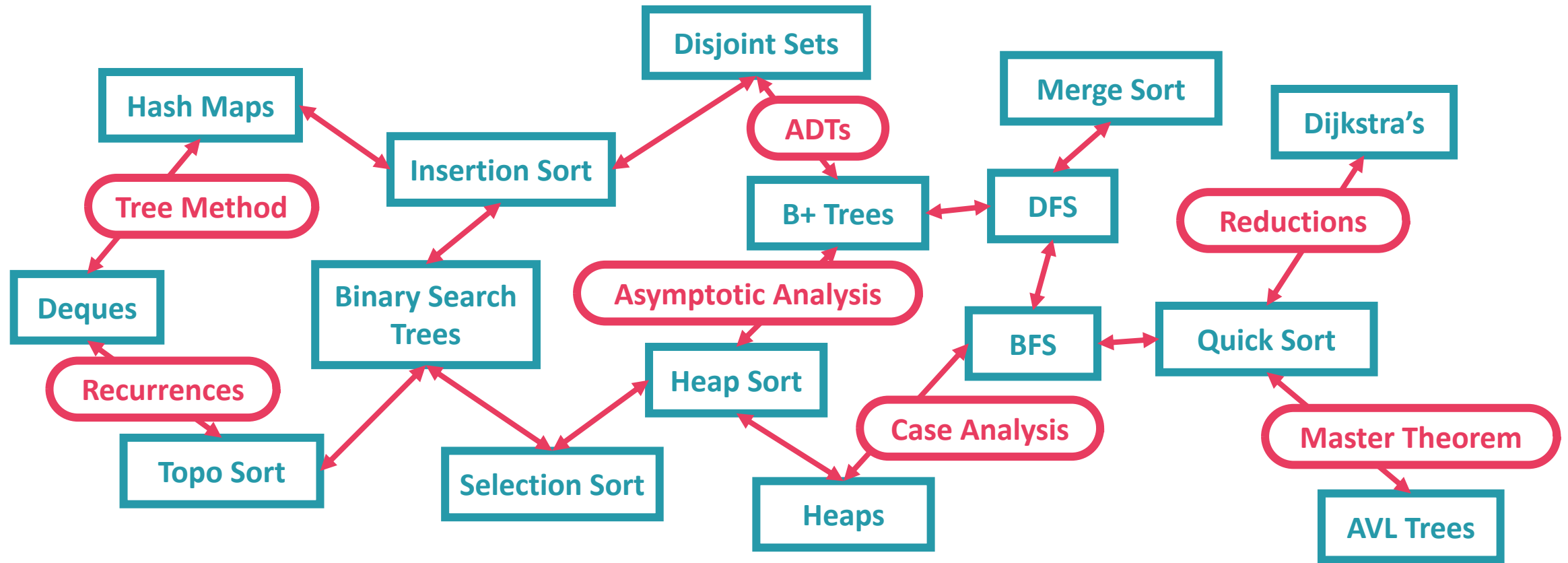
2. Pick up the vocabulary, skills, and practice needed to make **design decisions**. Learn to **evaluate** the tools in your CS toolbox



- Differences between technical implementations
- Evaluation can mean many different things!



# Data Structures & Algorithms



1. Build a **strong foundation** of data structures & algorithms
2. Learn to make **design decisions** and **evaluate** your tools

# Put Another Way...



1. Build a **strong foundation** of data structures & algorithms

=

Give you a cooler full of “tried-and-true” fish (that have been around for a while)



2. Learn to make **design decisions** and **evaluate** your tools

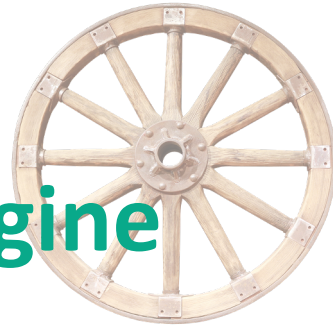
=

Teach you to fish... for your own ideas, data structures, and algorithms going forward



# ~~Reinventing the Wheel?~~

## *Studying the Wheel, So We Can Invent the Jet Engine*



P0

CSE 143 Review

- We spent a lot of time this quarter implementing our own data structures
  - But Java has a lot of these built-in. Why?

P1

Deque

1. Software engineering skills can't just be taught, they need to be **practiced**

P2

Maps

2. Not just writing code, but **analyzing & experimenting** with it

P3

Heap

3. Set you up to **invent** new, more complex data structures when the need arises

P4

Mazes

4. Understanding the **implementation** (the layer below) makes you more effective as the **client** (the layer above)!

## LEC 01

# Metacognition

- **Metacognition**: asking questions about your solution process.
- Examples:
  - **While debugging**: explain to yourself why you're making this change to your program.
  - **Before running your program**: make an explicit prediction of what you expect to see.
  - **When coding**: be aware when you're not making progress, so you can take a break or try a different strategy.
  - **When designing**:
    - Explain the tradeoffs with using a different data structure or algorithm.
    - If one or more requirements change, how would the solution change as a result?
    - Reflect on how you ruled out alternative ideas along the way to a solution.
  - **When studying**: what is the relationship of this topic to other ideas in the course?

# Learning to Bake in a CSE Class

- Think of what you've learned this quarter as a cookbook
  - ADTs are the chapters/category: Soups, Salads, Cookies, Cakes, etc
    - High-level descriptions of a category of functionality
    - You don't serve a soup when guests expect a cookie!
  - Data structures are the recipes: chocolate chip cookies, snickerdoodles, etc
    - Step-by-step, concrete descriptions of an item with specific characteristics
    - Understand your tradeoffs before replacing carrot cake with a wedding cake
- When you go out into the world ...
  - Figure out which category is required
  - Choose the specific recipe that best fits the situation
  - **Adapt or invent your own recipe when you need to!**



# What's Next?



**142**

## Introduction to Programming

- Methods
- Parameters, returns, values
- Conditionals
- Loops
- File Processing
- Arrays

**143**

## Object Oriented Programming

- Classes and Interfaces
- Recursion
- Linked lists and binary trees
- Sorting and Searching
- $O(n)$  analysis
- Generics

**373**

## Data Structures & Algorithms

- Design Decisions
- Data Structure Implementation
- Debugging & Testing
- Abstract Data Types
- Algorithmic Analysis
- Software Engineering



# What's Next?



In CSE 142 and 143, you learn **programming**. In 373, you open up the world of **computer science**.



And there's no shortage of places you can take the concepts from this class 😊

# If you want to...

- Learn about popular tools used in industry
  - e.g. Writing shell scripts in Linux, Using version control and git in more depth
- Understand exactly how your hardware and code interact at a very low level inside a physical computer
  - Want to go more than one lecture deep to learn how memory actually works?

# Consider taking...

**CSE 374**

Intermediate Programming Concepts and Tools

- Offered 20au (Champion), 21wi (Wilcox), 21sp (Hazen)

# If you want to...

- Learn new strategies for problem-solving through programming languages that look completely different from Java
- Learn how programming languages actually work
  - What happens starting the moment you click “run” in IntelliJ?
  - What does the Java compiler even do?

# Consider taking...

**CSE 413**

Programming Languages

- Offered 21sp (Perkins)

# If you want to...

- Level up the scope of the applications you can work on by learning how to store, access, and query huge amounts of data
  - *So many* projects in industry incorporate a database
  - Heard of big data™? You'll want a bit more than an in-memory hash map
- Learn about parallelism and concurrency

# Consider taking...

**CSE 414**

Database Systems

- Offered 20au (Maas), 21wi (Maas), 21sp (Thompson)



# If you want to...

- Use those databases full to the brim with big data™ to do machine learning on new and interesting problems
- Understand *why* machine learning has so much potential to overturn entire industries
  - And in the process, get practical experience working on ML projects with real datasets

# Consider taking...

**CSE 416**

Machine Learning

- Offered 21sp (Schafer)

# If you want to...

- Learn more about the underlying theory behind data structures and algorithms covered in this class
- Learn about designing complex algorithms
- Understand the *limits* of computation and the fundamental problems that remain unsolved in computer science (P vs. NP, anyone?)

# Consider taking...

**CSE 417**

Algorithms and Computational Complexity

- Offered 20au (Anderson), 21wi (Weber)

# If you want to...

- Build a website or web app
  - Either the frontend (what visitors see in their browser) or the backend (what runs on the server to compute data)
- Learn the fundamentals of a number of web technologies that make it easier for you to learn more on your own

# Consider taking...

**CSE 154, INFO 343, or INFO 344**

- CSE 154 offered 20au (Gibbon), 21sp (Gibbon)

# Your Next Language

- Java is our lens, but these concepts generalize to any language!
  - Learning your second language is *much* easier than learning your first 😊
- Mainstream languages you might be interested in:
  - Python (“pseudocode that runs”)
  - C# (“Java done right”)
  - HTML/CSS/JavaScript (“building blocks of the web”)
  - C/C++ (“if you want full control”)
- Or, consider an unconventional/up-and-coming one:
  - Haskell
  - Racket
  - Rust
  - Prolog

# Learning New Things

- Nothing in Computer Science is out of your grasp!
  - This class is designed to give you the foundation to go out and
- Where to next?
  - You can find tutorials on almost any topic via google
  - Coursera: lots of good online courses
  - Google “open source CS curriculum” or “what every CS major should know” if you want a more curated list
  - Try contributing to open source
  - Try attending hackathons!
  - StackOverflow: pick interesting tags, sort by top, and read
- Advice: Take charge of your own education, prioritize practice over passive learning, be persistent, and always let your curiosity lead



# These Unprecedented Times

- As much as we want to focus on making 373 an excellent experience, we can't forget what's going on in the world around us
- I know online learning hasn't been easy for anyone
  - Thank you for letting us know your experiences and feedback
  - Thank you for reaching out to ask for help when you needed it
  - Thank you for being so understanding when my Zoom cuts out on the reg
- As a course staff, we are blown away by how much you've engaged with the course despite the fact that it all happens in a rectangle on your computer screen. **Thank you.**

**AMA**  
**ASK ME ANYTHING!**





**Thanks for  
a great quarter!**

– The CSE 373 20SU Staff ☺