Lecture 28: Real-World Software Engineering

CS 2110

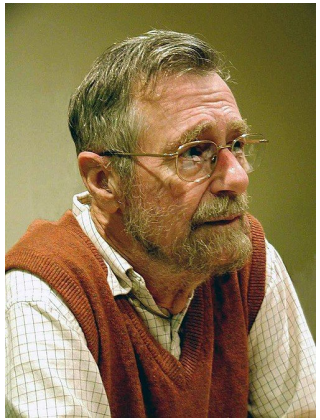
May 5, 2026

BLASTing Genes

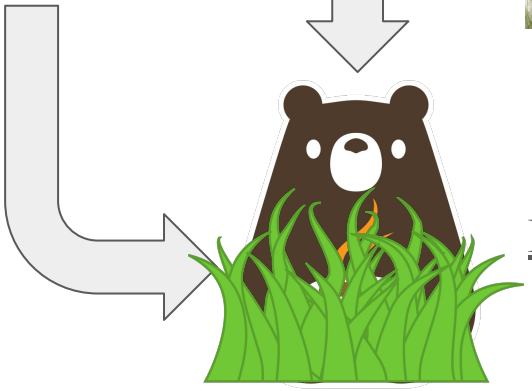
Ayman Abou-Alfa

Let's say we are *biologists*…

Ursus arctos



Homo sapiens



Java Bear

Which species did the Java Bear evolve from?

<u>Physical characteristics:</u>

1) Looks like a brown bear
2) Great at object-oriented programming

Poll Everywhere

PollEv.com/javabear text **javabear** to 22333

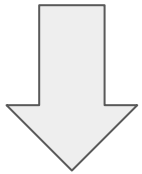
JavaGene: ATTGTAGCCGATTACG

Given a database of known genes, how should we structure our database to quickly find JavaGene matches?

- HashMap **(A)**
- HashSet **(B)**
- LinkedList **(C)**
- Heap **(D)**

Turns out…there are *no exact* matches! Why?

The Java Bear evolved from another species!

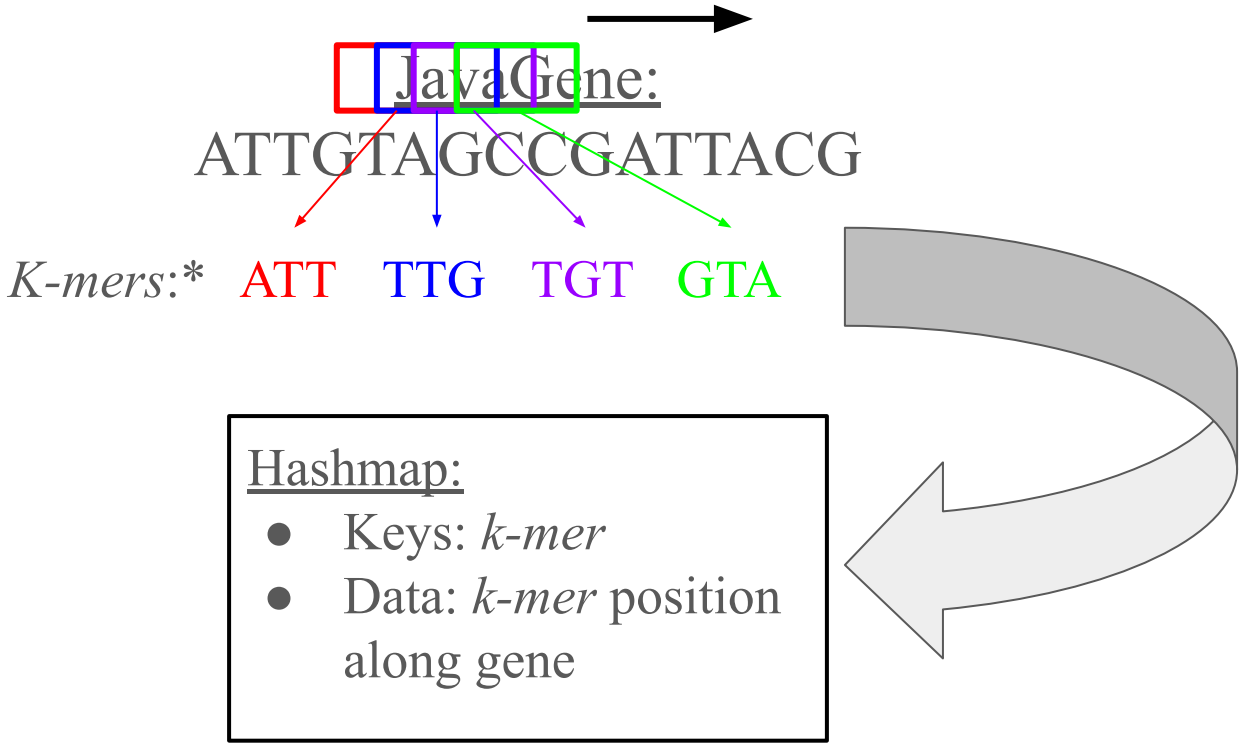


The JavaGene accumulated **mutations** (small chemical changes) during evolution!

Better question:

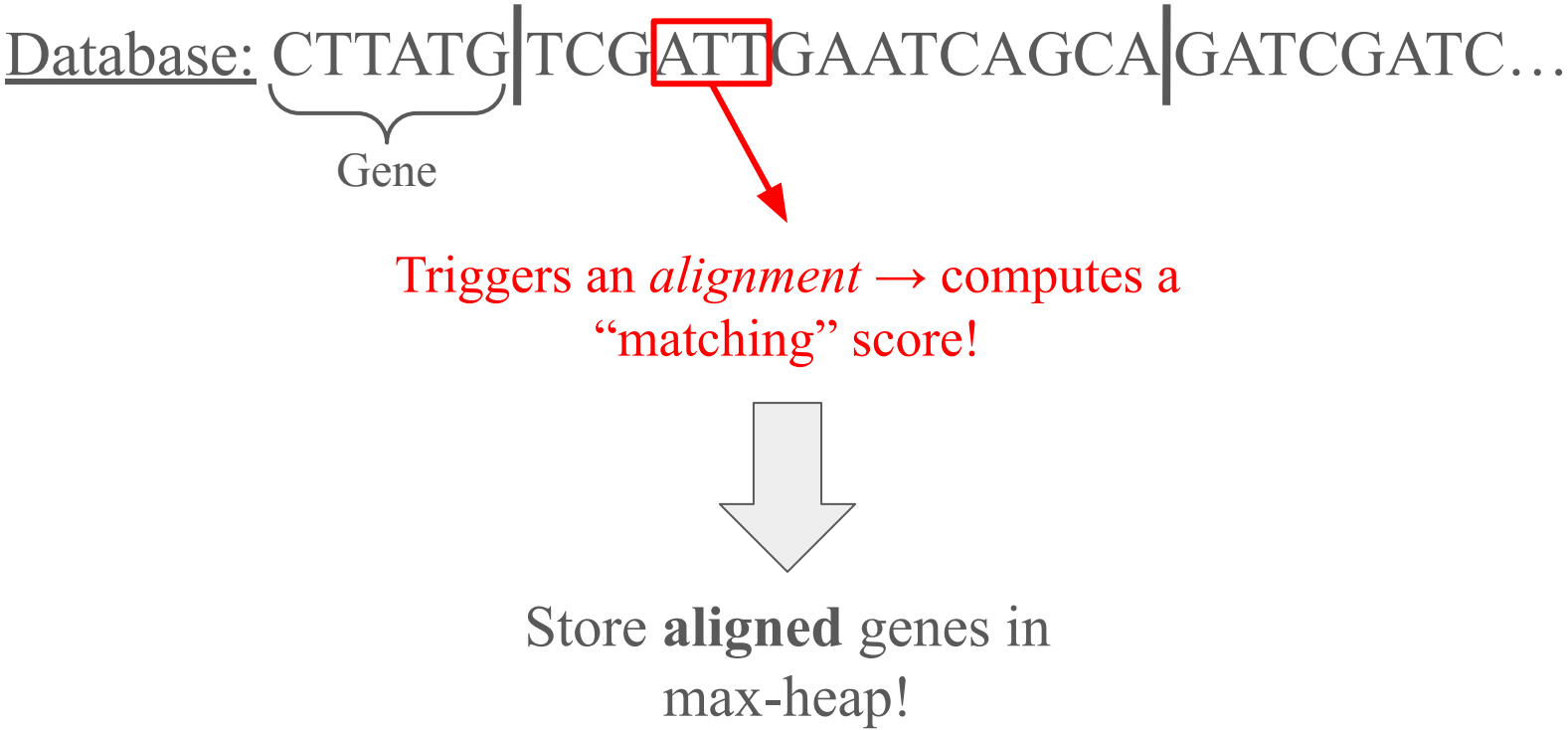
How do we find the *best-possible* genetic matches, which contain the fewest mutations?

Instead of a direct approach, let's *break our gene apart*…



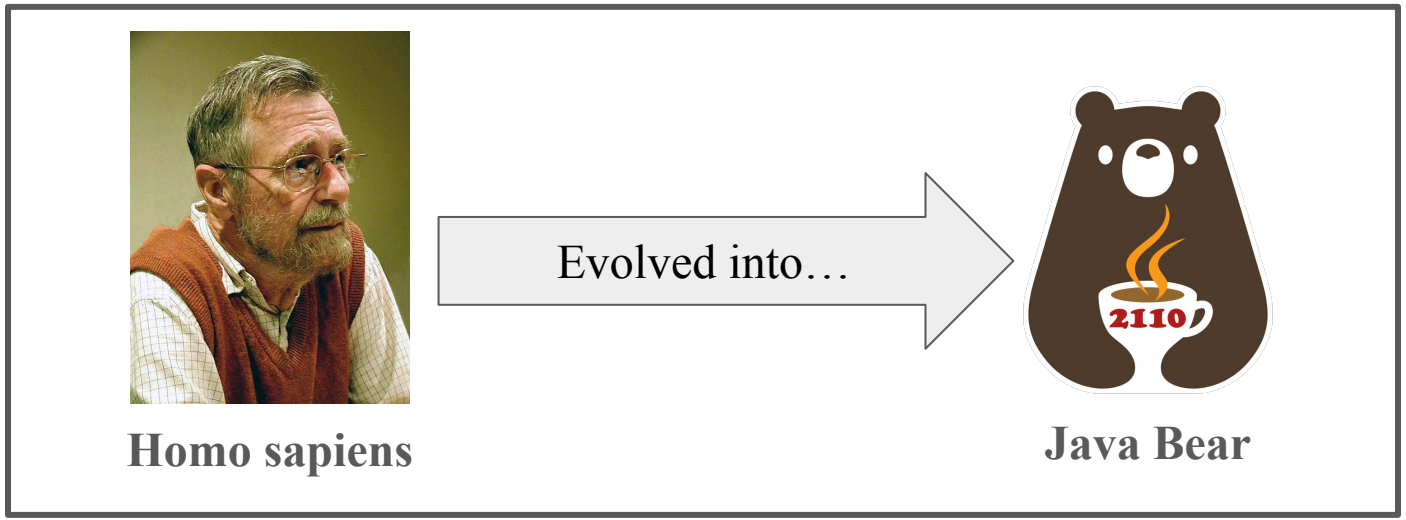
*in practice, these are longer segments of the gene

Linearly scan our database, searching for *k-mers*:



B(asic) L(ocal) A(lignment) S(earch) T(ool), Altschul et al.

Retrieving elements from the **max-heap**. → Our genes, ranked from *best to worst*!

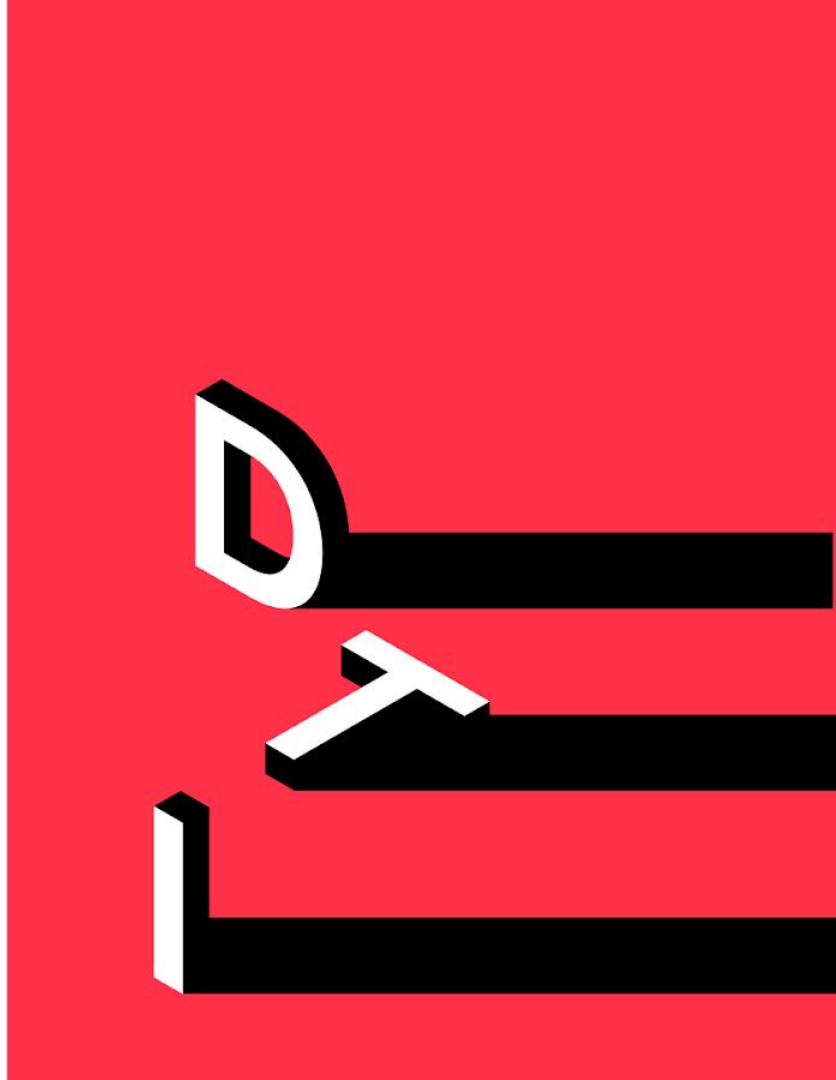


CS 2110 x Web Development (Digital Tech and Innovation)

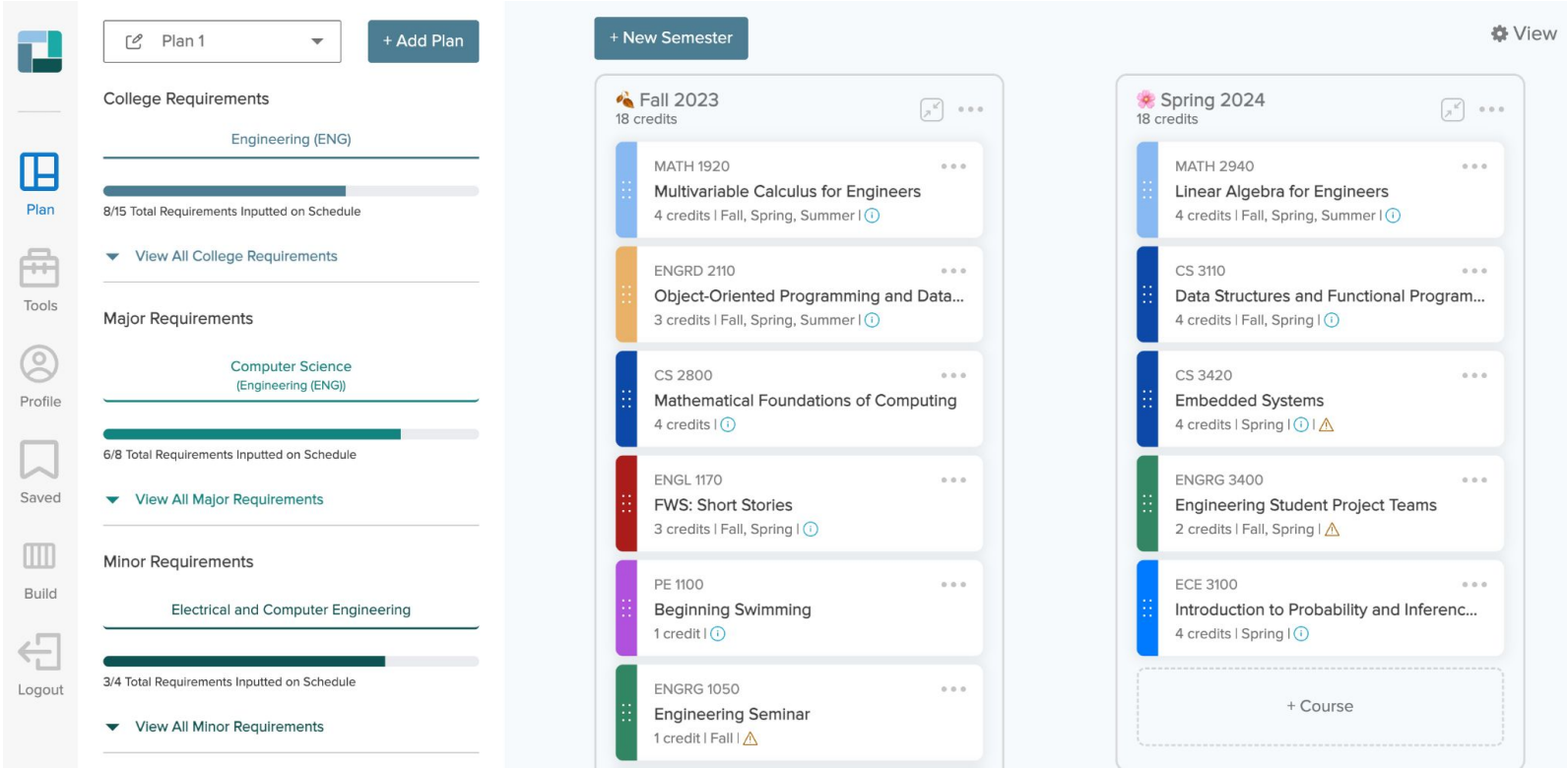
Chris and Ella

What is DTI?

DTI is a software engineering project team that **creates technology for community impact.**



CoursePlan


Plan 1
+ Add Plan
College Requirements
Engineering (ENG)
8/15 Total Requirements Inputted on Schedule
View All College Requirements
Major Requirements
Computer Science
(Engineering (ENG))
6/8 Total Requirements Inputted on Schedule
View All Major Requirements
Minor Requirements
Electrical and Computer Engineering
3/4 Total Requirements Inputted on Schedule
View All Minor Requirements
Plan
Tools
Profile
Saved
Build
Logout
+ New Semester
View
Fall 2023
18 credits
MATH 1920
Multivariable Calculus for Engineers
4 credits | Fall, Spring, Summer |
ENGRD 2110
Object-Oriented Programming and Data...
3 credits | Fall, Spring, Summer |
CS 2800
Mathematical Foundations of Computing
4 credits |
ENGL 1170
FWS: Short Stories
3 credits | Fall, Spring |
PE 1100
Beginning Swimming
1 credit |
ENGRG 1050
Engineering Seminar
1 credit | Fall |
Spring 2024
18 credits
MATH 2940
Linear Algebra for Engineers
4 credits | Fall, Spring, Summer |
CS 3110
Data Structures and Functional Program...
4 credits | Fall, Spring |
CS 3420
Embedded Systems
4 credits | Spring |
ENGRG 3400
Engineering Student Project Teams
2 credits | Fall, Spring |
ECE 3100
Introduction to Probability and Inferenc...
4 credits | Spring |
+ Course

How do we build large software?

You've...

- built projects in small teams
- ran your code against our autograders
- probably already forgotten what you wrote a few months ago

When you build large software, you...

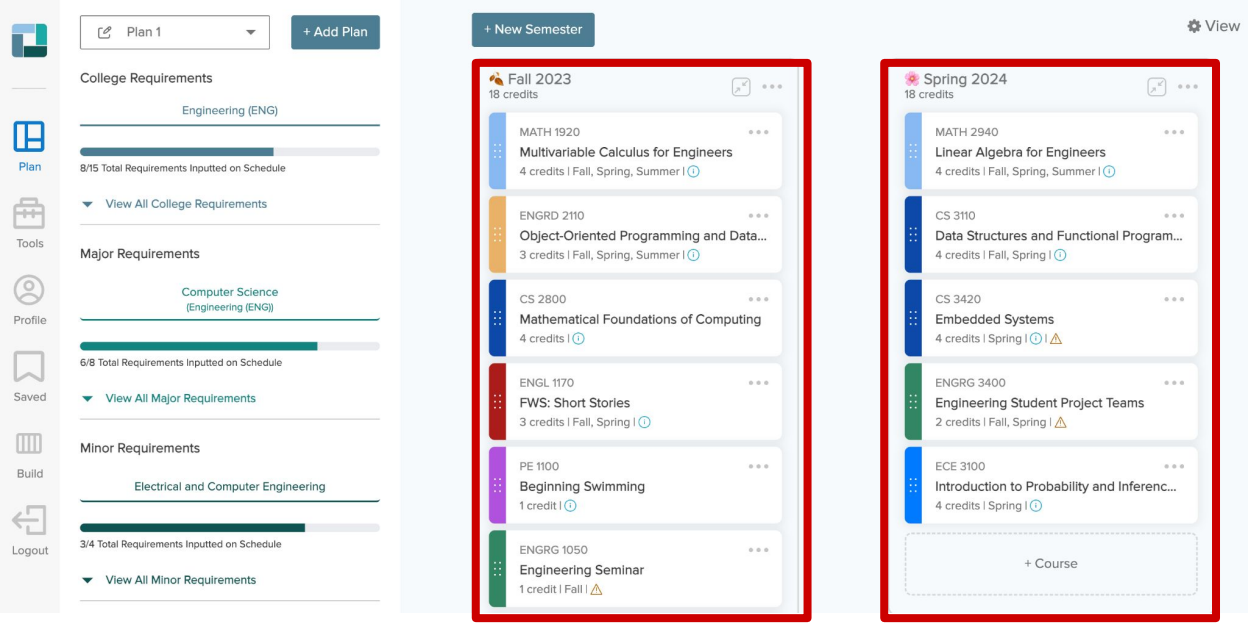
- work with not only several developers, but also designers, product managers, marketers, etc.
- have no rubric or office hours
- build on code that has been written for years before you

Specifications and Documentation

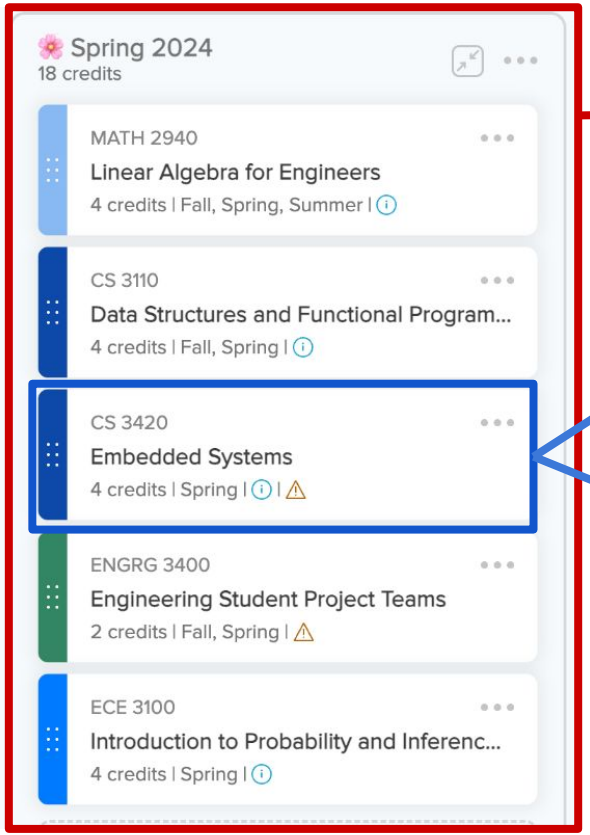
```
/** Reachable at POST /api/courses/get-by-info
 * @body number: a course's number
 * @body subject: a course's subject code
 * Gets a course by its subject and number
 */
courseRouter.post('/get-by-info', async (req, res) => {
  try {
    const { number, subject }: CourseInfoRequestType = req.body;
    const course = await getCourseByInfo({ number, subject });

    if (!course) {
      return res.status(404).json({
        error: `Course could not be found with subject: ${subject} and number: ${number}`
      });
    }

    return res.status(200).json({ result: course });
  } catch (err) {
    return res
      .status(500)
      .json({ error: `Internal Server Error: ${err.message}` });
  }
});
```

Abstraction and Encapsulation

Components are the building blocks of a frontend application.



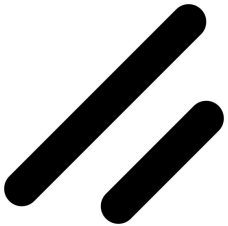
Abstraction and Encapsulation




```
private List<Course>;
private String semesterName;
private boolean hidden;
```

```
private String courseName;
private int credits;
```

Outside Libraries



Testing

Unit Testing: testing the smallest parts of an application.

Integration Testing: testing the medium-est parts of an application

End-to-end testing: testing the whole application

```
describe('Course functionality unit tests', () => {
  test('Getting review of course that exists', async () => {
    const res = await axios.post(
      `http://localhost:${testPort}/api/courses/get-reviews`,
      { courseId: 'oH37S3mJ4eAsktypy' }
    );

    const reviews = await Reviews.find({
      class: 'oH37S3mJ4eAsktypy',
      reported: 0,
      visible: 1
    });
    expect(res.data.result.length).toBe(reviews.length);

    const classOfReviews = reviews.map((r) => r.class);
    expect(res.data.result.map((r) => r.class).sort()).toEqual(
      classOfReviews.sort()
    );
  });
```

Code Reviews

frontend/src/components/Admin/TeamEvent/TeamEventDashboard.tsx Outdated

Comment on lines +76 to +83

```
76  +  const getPeriodIndex = (date: Date): number => {
77  +    for (let i = 0; i < TEC_DEADLINES.length; i += 1) {
78  +      if (date <= TEC_DEADLINES[i]) {
79  +        return i;
80  +      }
81  +    }
82  +    return TEC_DEADLINES.length - 1;
83  +  };
```

cchrischen on Mar 19, 2025 Member

This logic looks very similar to `getTECPeriod` defined below. Can we refactor to use that function you defined in your last PR?

1

Reply...

Resolve conversation

Takeaways

+1.0

All other helper methods include correct and sufficient documentation of their specs.

+1.0

All helper methods are `private` and include complete JavaDoc specifications

+0.5

Partial credit: Helper methods include incomplete specifications (see comments) or are fully specified but not `private`

These rubric items that may seem annoying or tedious are actually one of the most important aspects of developing software in the real world!

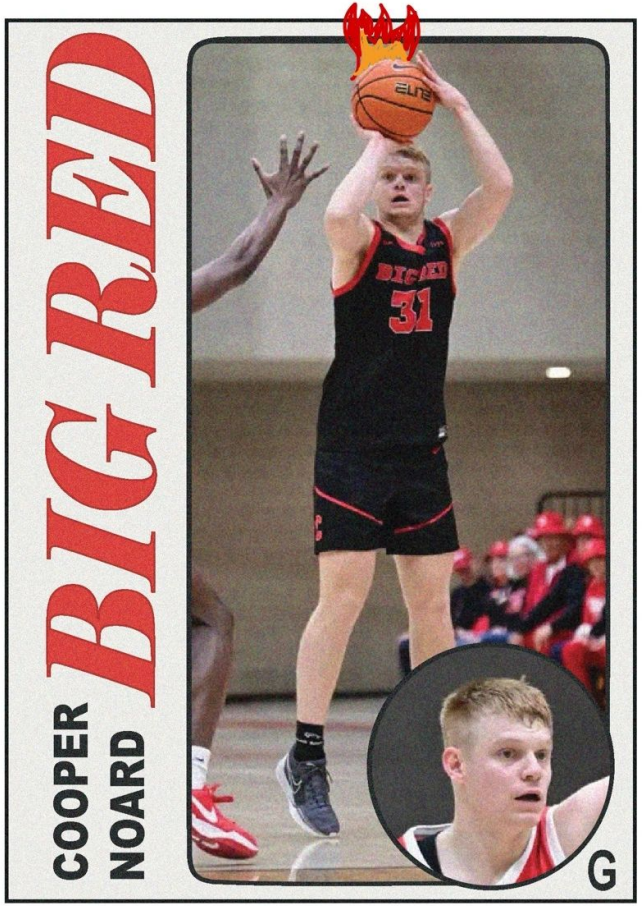
2110 and Behavioral Economics
“The Hot Hand”

Nikhil Chinchalkar

QUESTION

Does the “hot hand” effect exist?

“if a person made their last shot, they are more likely to make the next one…”



Photos by Caroline Sherman (Cornell University Athletics)

NOTES

- Notation: $P(H|H) > P(H) > P(H|M)$
- Coined by a Cornell professor in 1985
- Statistically proven to be a fallacy…
- … but there was an error in the original paper
- I redid the analysis as a project for a club
  - nikhilc52.github.io/hot_hand/website/

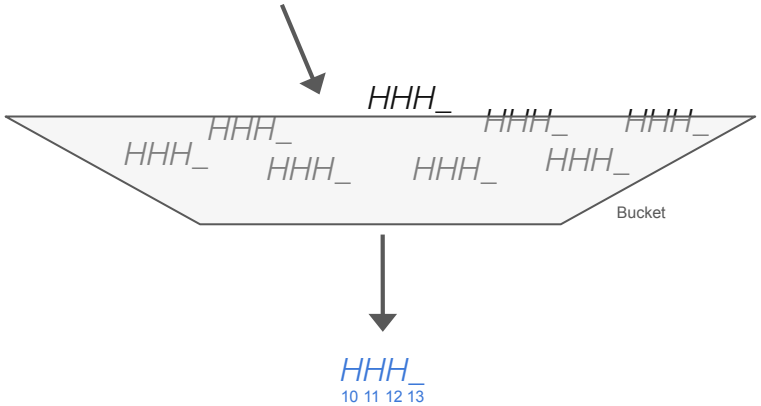
CALCULATION OF P(H|HHH)

ERROR

Imagine you took a string of H’s and M’s from a truly independent 50% shooter…

HMHHMMH…MHHMHMH

…from there you collected all HHH_ sequences…



…and drew one at random…

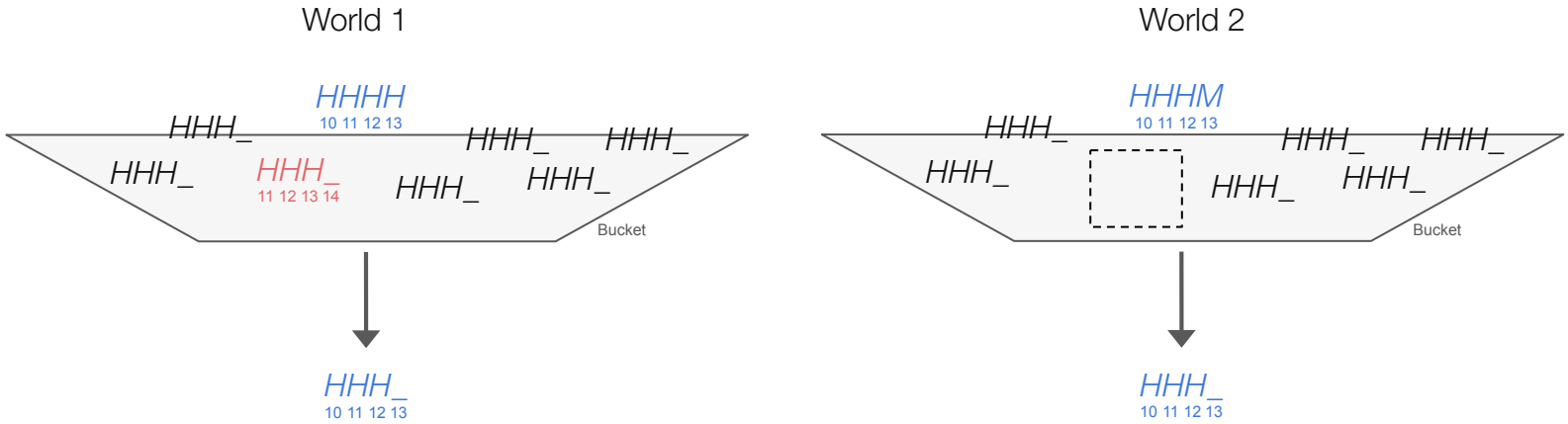
…what would the probability of seeing an “H” at the “_” position be?

If “_” was “H”, then we’d also have another sequence in the bucket: *HHH_* 11 12 13 14

So the probability of selecting our original *HHH_* 10 11 12 13 would be slightly lower…

CALCULATION OF P(H|HHH)

ERROR



Given we selected HHH_ (10 11 12 13) …

…we’re more likely to be in World 2 (compared to World 1)...

…so it's more likely “_” is an “M” - even if H’s and M’s are independent!

This means the paper severely *underestimated* the “hot hand”

If this is confusing, don’t worry: it took the best psychologists and economists ~30 years to realize

BIAS CALCULATION

CALCUALTION

So the measurement is biased…

…how do we correct it?

Algorithm 1: Recursive formula that builds the collection of dictionaries D. Of interest are the dictionaries $D(0,n)$ for $n = k+1,\ldots,N$ which correspond to the joint distribution of the total number of (successes, failures) that immediately follow k consecutive successes in n trials.

Function `Count_Distribution`$(N,\ k,\ p)$**:**

`/* For the definition of` $D(\ell,r)$, $A^{m':p'}$ `and` $A \uplus B$ `below, see text. */`

$q \leftarrow 1-p$
for $n = 0,\ldots,N$ **do**
  $L \leftarrow \min\{k,n\}$
  for $\ell = L,\ldots,0$ **do**
    $r \leftarrow n-\ell$
    if $r = 0$ **then**
      $D(\ell,r) \leftarrow ((0,0):1)$
    else if $r > 0$ **then**
      if $\ell < k$ **then**
        $D(\ell,r) \leftarrow D(0,r-1)^{(0,0):q} \uplus D(\ell+1,r-1)^{(0,0):p}$
      else if $\ell = k$ **then**
        $D(\ell,r) \leftarrow D(0,r-1)^{(1,0):q} \uplus D(k,r-1)^{(0,1):p}$
  end
end
return D

2110 Concepts

- Recursion
- Dictionaries/ `Maps`
- `Sets`

2800 Concepts

- Conditional Probability
- Joint Distributions
- Independence

CS 2110

TAKEAWAY

The authors of these papers (and me) are psychologists and behavioral economists, **not computer scientists:**

Thomas Gilovich
Psychologist

Amos Tversky
Cognitive Psychologist

Robert Vallone
Psychologist

Adam Sanjurjo
Behavioral Economist

Joshua Miller
Decision Scientist

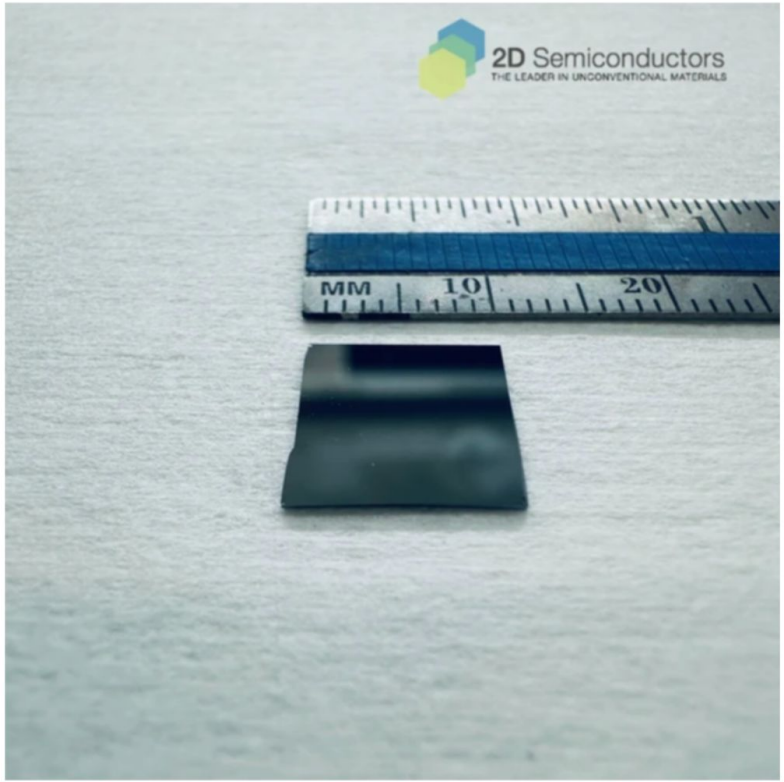
But CS 2110 is so foundational that even the most tangential concepts use 2110 principles…

No matter what you do with your life, a piece of CS 2110 will always be with you!

Big O Notation and Superconductors

Brian Xia

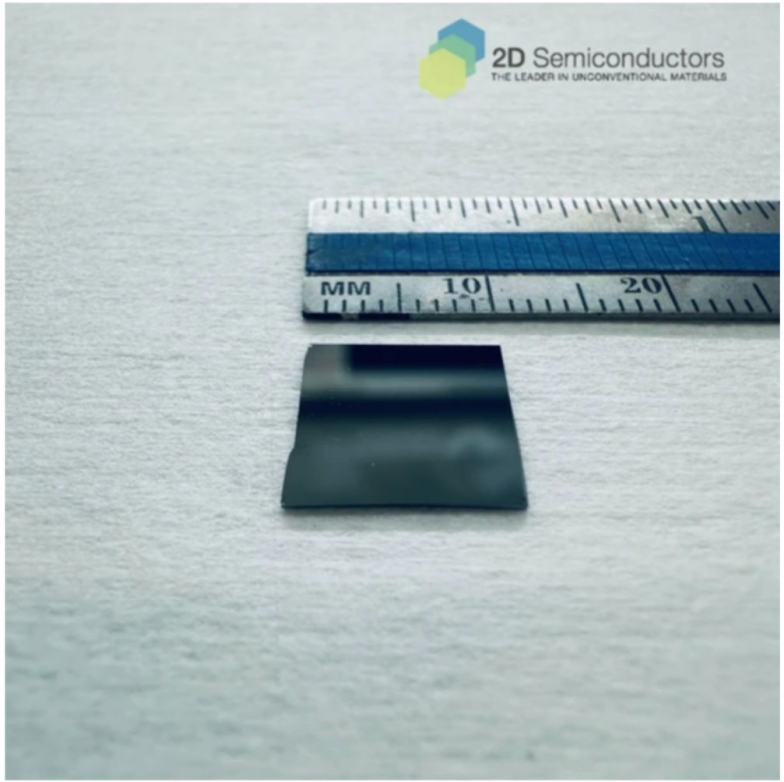
Superconductors



Runs electricity with 0 resistance when really cold

Make superconducting thin films for
- Nanodevices
- Quantum computer circuits

Superconductors



NbTiN thin film

SKU: PLD-NbTiN

$590.00

Starting at $31.79/mo or as low as 0% APR with PayPal. Learn more

QUANTITY:

1

ADD TO CART

ADD TO QUOTE

*Also takes a long time to make, and easy to mess up

Where research and runtime come in

- Machine learning models can estimate results in advance
  - Research Goal: Find some algorithm that best fits the superconductor scenario
- Personal story about this algorithm
  - Has to deal with runtime and Big O analysis

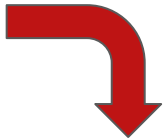
Algorithm 1 Reptile (serial version)

Initialize ϕ, the vector of initial parameters
for iteration $= 1, 2, \ldots$ **do**
  Sample task τ, corresponding to loss L_τ on weight vectors $\widetilde{\phi}$
  Compute $\widetilde{\phi} = U_\tau^k(\phi)$, denoting k steps of SGD or Adam
  Update $\phi \leftarrow \phi + \epsilon(\widetilde{\phi} - \phi)$
end for

One of the algorithms we tried. You don't need to focus on the exact steps, just know it's designed to be fast

Algorithm 1 Reptile (serial version)

Initialize ϕ, the vector of initial parameters
for iteration $= 1, 2, \ldots$ **do**
  Sample task τ, corresponding to loss L_τ on weight vectors $\widetilde{\phi}$
  Compute $\widetilde{\phi} = U_\tau^k(\phi)$, denoting k steps of SGD or Adam
  Update $\phi \leftarrow \phi + \epsilon(\widetilde{\phi} - \phi)$
end for



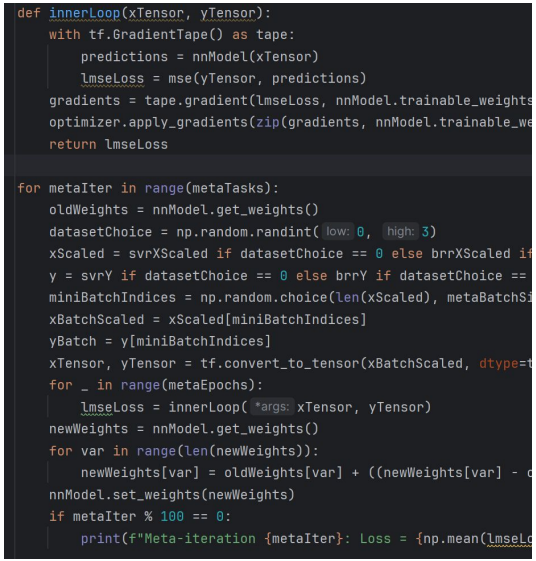
```python
def innerLoop(xTensor, yTensor):
    with tf.GradientTape() as tape:
        predictions = nnModel(xTensor)
        lmseLoss = mse(yTensor, predictions)
    gradients = tape.gradient(lmseLoss, nnModel.trainable_weights)
    optimizer.apply_gradients(zip(gradients, nnModel.trainable_weights))
    return lmseLoss

for metaIter in range(metaTasks):
    oldWeights = nnModel.get_weights()
    datasetChoice = np.random.randint( low: 0,  high: 3)
    xScaled = svrXScaled if datasetChoice == 0 else brrXScaled if datasetChoice == 1 else gprXScaled
    y = svrY if datasetChoice == 0 else brrY if datasetChoice == 1 else gprY
    miniBatchIndices = np.random.choice(len(xScaled), metaBatchSize, replace=False)
    xBatchScaled = xScaled[miniBatchIndices]
    yBatch = y[miniBatchIndices]
    xTensor, yTensor = tf.convert_to_tensor(xBatchScaled, dtype=tf.float32), tf.convert_to_tensor(yBatch, dtype=tf.float32)
    for _ in range(metaEpochs):
        lmseLoss = innerLoop( *args: xTensor, yTensor)
    newWeights = nnModel.get_weights()
    for var in range(len(newWeights)):
        newWeights[var] = oldWeights[var] + ((newWeights[var] - oldWeights[var]) * metaStepSize)
    nnModel.set_weights(newWeights)
    if metaIter % 100 == 0:
        print(f"Meta-iteration {metaIter}: Loss = {np.mean(lmseLoss.numpy()):.6f}")
```

The algorithm looks like this in code

Runtime of this algorithm?

- Involves around three nested loops
  - O(n * m * k) time, with placeholder variables for each loop
- In total, ~400k iterations of constant work

How much time does it take to run?



```
def innerLoop(xTensor, yTensor):
    with tf.GradientTape() as tape:
        predictions = nnModel(xTensor)
        lmseLoss = mse(yTensor, predictions)
    gradients = tape.gradient(lmseLoss, nnModel.trainable_weights
    optimizer.apply_gradients(zip(gradients, nnModel.trainable_we
    return lmseLoss

for metaIter in range(metaTasks):
    oldWeights = nnModel.get_weights()
    datasetChoice = np.random.randint( low: 0,  high: 3)
    xScaled = svrXScaled if datasetChoice == 0 else brrXScaled if
    y = svrY if datasetChoice == 0 else brrY if datasetChoice ==
    miniBatchIndices = np.random.choice(len(xScaled), metaBatchSi
    xBatchScaled = xScaled[miniBatchIndices]
    yBatch = y[miniBatchIndices]
    xTensor, yTensor = tf.convert_to_tensor(xBatchScaled, dtype=t
    for _ in range(metaEpochs):
        lmseLoss = innerLoop( *args: xTensor, yTensor)
    newWeights = nnModel.get_weights()
    for var in range(len(newWeights)):
        newWeights[var] = oldWeights[var] + ((newWeights[var] - o
    nnModel.set_weights(newWeights)
    if metaIter % 100 == 0:
        print(f"Meta-iteration {metaIter}: Loss = {np.mean(lmseLo
```

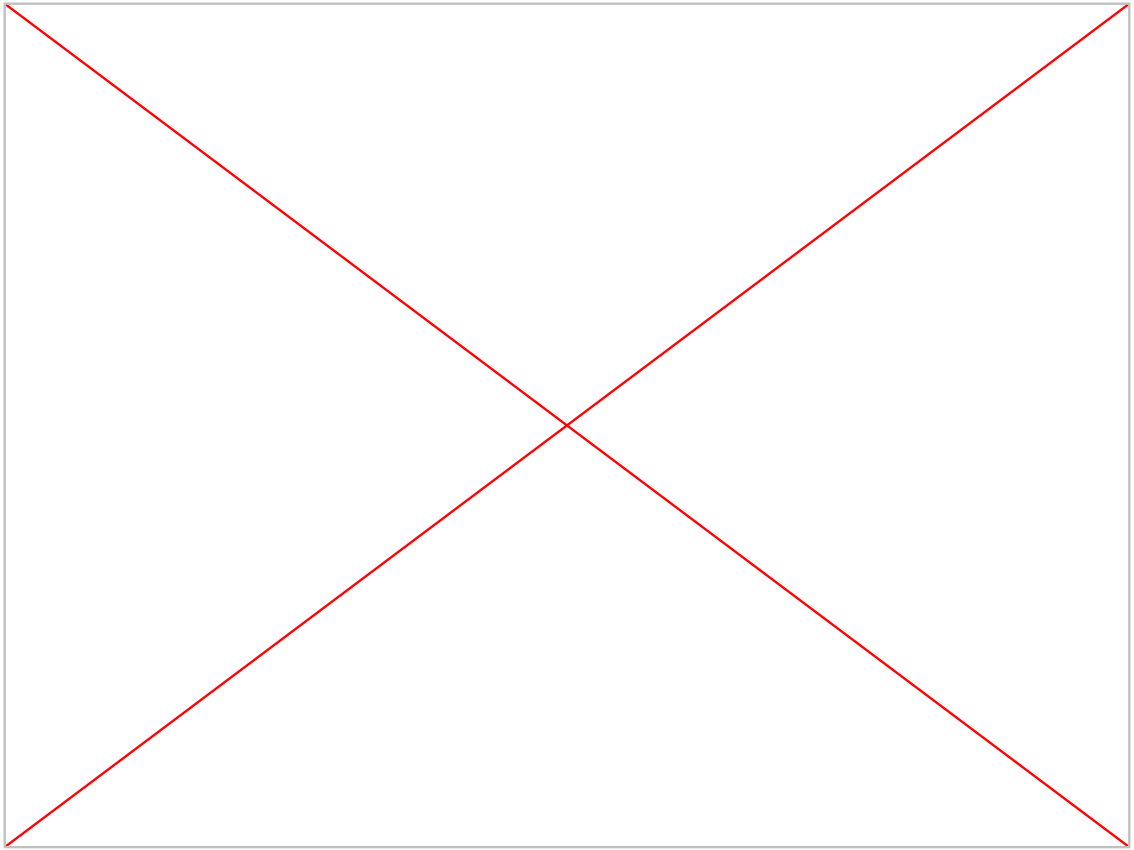
Poll Everywhere

PollEv.com/javabear text **javabear** to 22333

Take a guess: roughly how long would that code take to execute? It has 3 nested loops, doing ~400k constant operations.

5 seconds	**(A)**
1 minute	**(B)**
5 minutes	**(C)**
10 hours	**(D)**

Video of training the model



Poll Everywhere

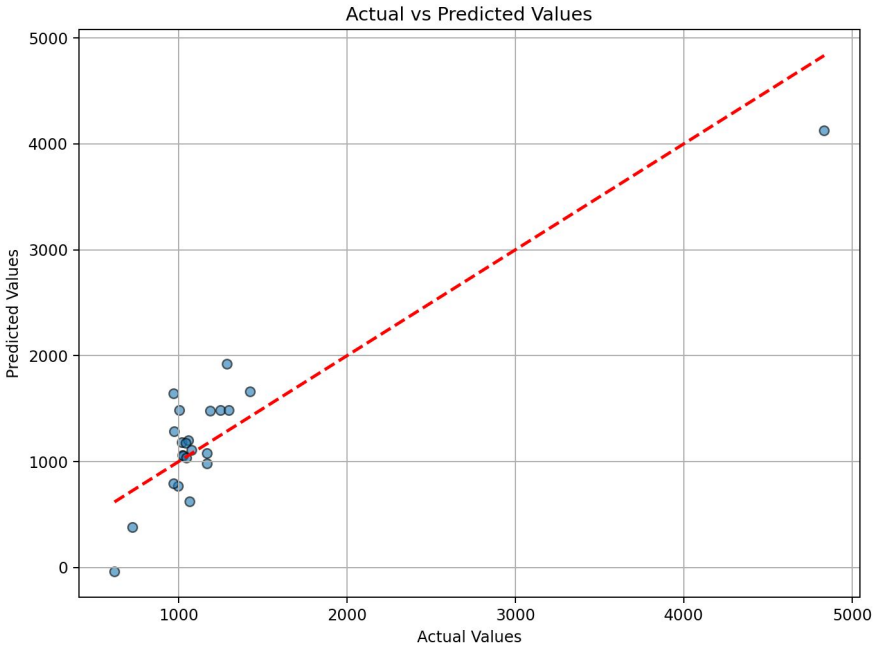
PollEv.com/javabear text **javabear** to 22333

Take a guess: roughly how long would that code take to execute? It has 3 nested loops, doing ~400k constant operations.

- 5 seconds **(A)**
- 1 minute **(B)**
- 5 minutes **(C)**
- 10 hours **(D)**

Very Long Runtime?

- This “fast” algorithm should not be taking 10 hours
  - It at least produced models that looked good
- Big O doesn’t line up with runtime
  - O(n * m * k) shouldn’t need 10 hours



The model made from the algorithm at least worked

What happened?

- Something to know about Tensorflow
  - C++ does heavy computations behind the scenes
  - Python sends operations to C++
- Crossing the boundary on every operation adds **constant** overhead
  - Not included in Big O
- Can reduce number of crossings

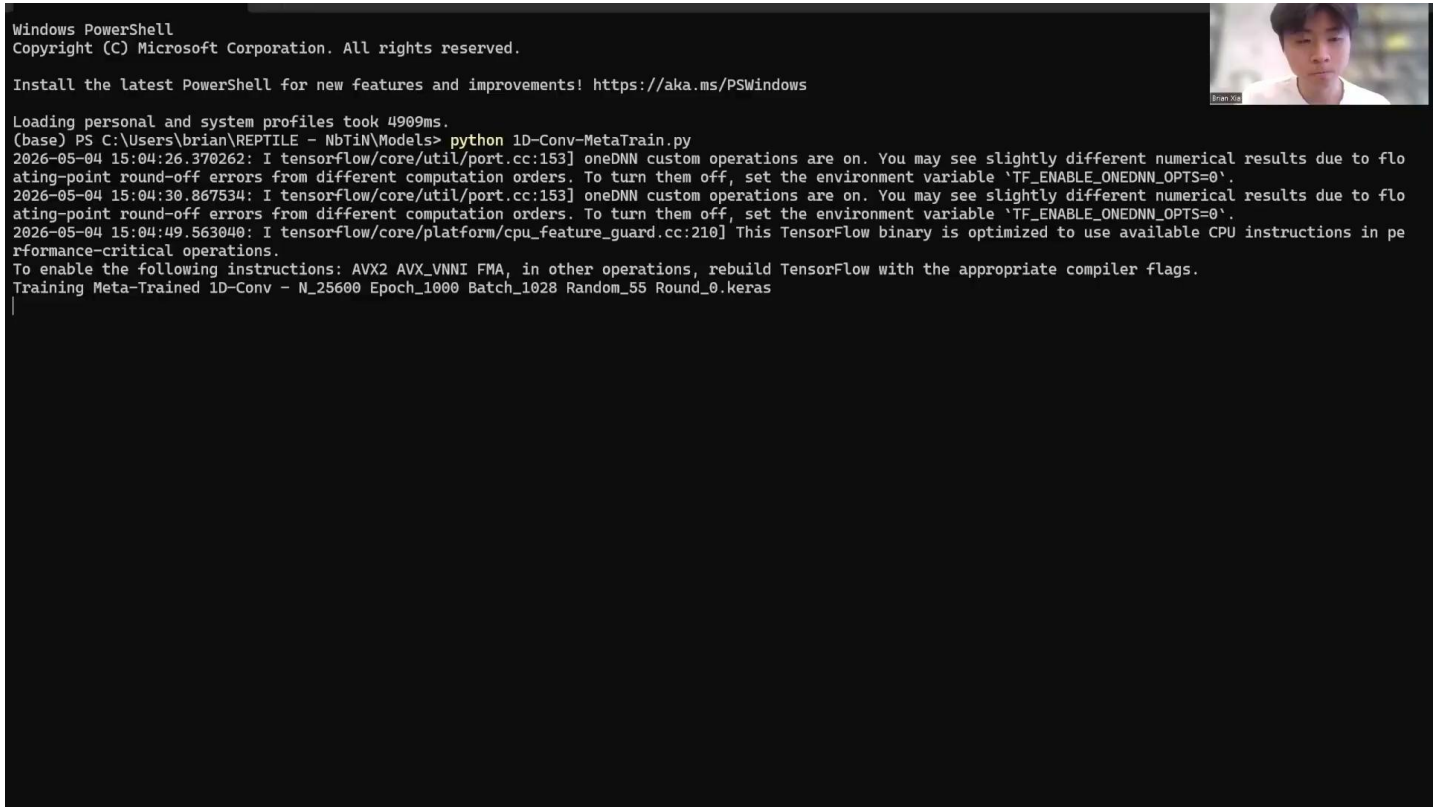
Before

```
def innerLoop(xTensor, yTensor):
    with tf.GradientTape() as tape:
        predictions = nnModel(xTensor)
        lmseLoss = mse(yTensor, predictions)
    gradients = tape.gradient(lmseLoss, nnModel.trainable_weights)
    optimizer.apply_gradients(zip(gradients, nnModel.trainable_weights))
    return lmseLoss
```

After

```
@tf.function  1 usage
def innerLoop(xTensor, yTensor):
    with tf.GradientTape() as tape:
        predictions = nnModel(xTensor)
        lmseLoss = mse(yTensor, predictions)
    gradients = tape.gradient(lmseLoss, nnModel.trainable_weights)
    optimizer.apply_gradients(zip(gradients, nnModel.trainable_weights))
    return lmseLoss
```

Video of training the model (with this change)

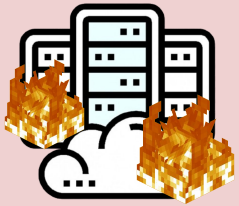


Lessons Learned

- Asymptotic complexity != actual runtime
  - Big O ignores constants
- Fundamentals always point in the right direction
  - Saves you a lot of time

https://github.com/brianxia273/NNReactiveSputtering
https://doi.org/10.1016/j.jmsy.2022.02.004.

https://github.com/brianxia273/NbTiN-NNReactiveSputtering



CS 2110 x Cornell Data Science (CDS)

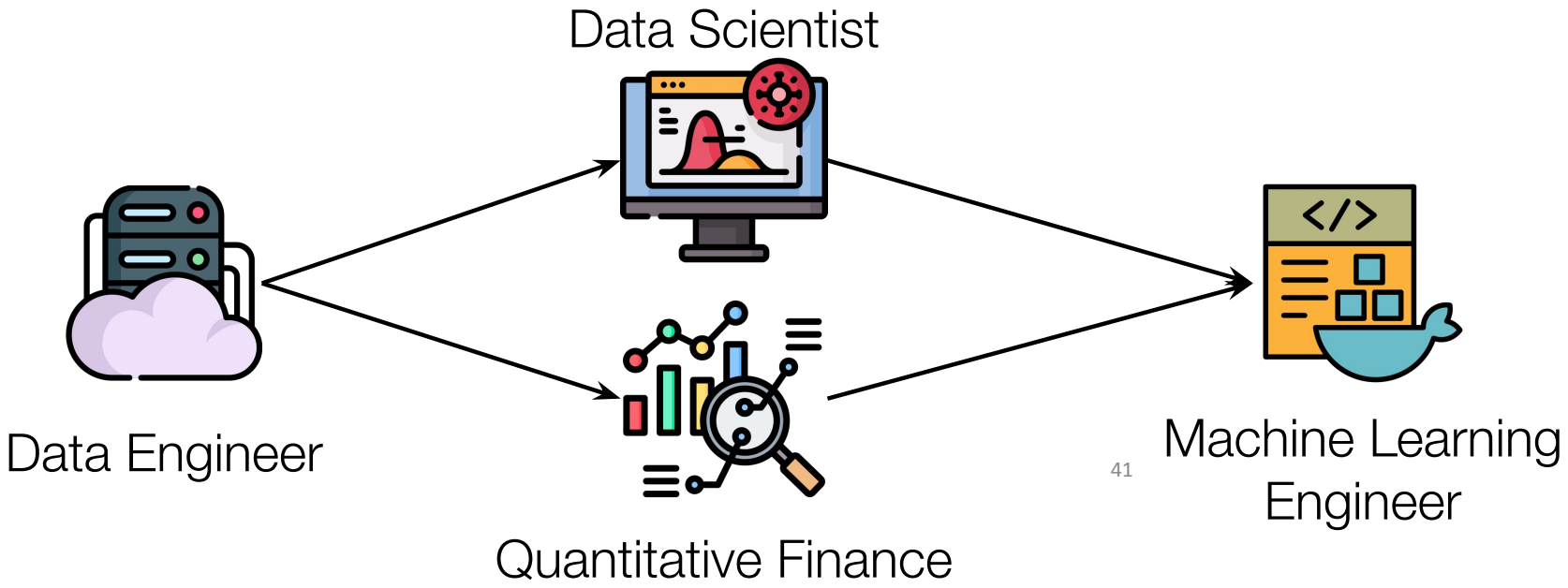
Naijei Jiang

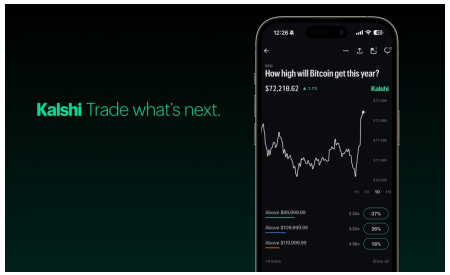
Who are we?

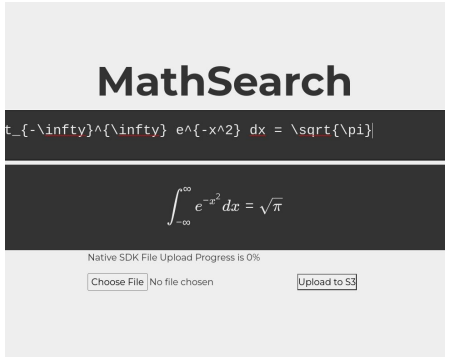
Cornell Data Science (CDS) is an undergraduate project team focused on building **data-driven** solutions to **real-world** problems.

- 70+ members
- 4 subteams
- 4+ colleges
- 1 community

Our subteams







Our Projects

Compute Cluster: High-performance computing environment for projects.

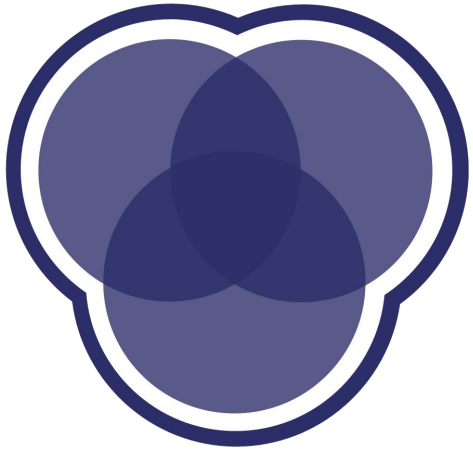
Geometry Dash RL Bot: Training an RL bot to master Geometry Dash levels efficiently

Smash AI: AlphaGo-like RL agent to play Super Smash Bros

Kalshi Trading Platform: Find correlated Kalshi markets for low-risk arbitrage trading

Proposed by members like YOU each semester.

As well as Corporate Projects!



Distributed Downloader

99% Downloaded…

Overview

- Slow downloading times?
  - Even if the people around you have the files?

- Distributed Downloader!
  - Inspired by BitTorrent™

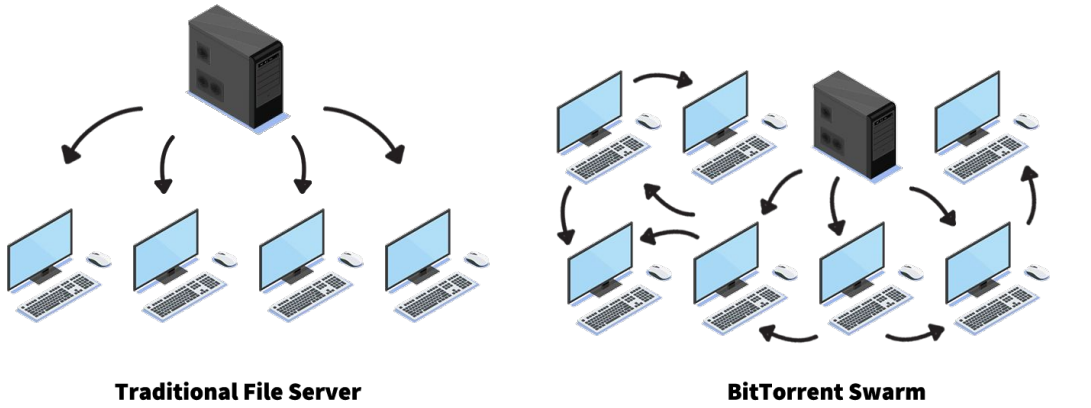
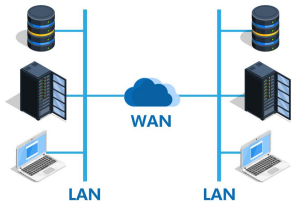


Image from TechTerms.com

Data

Environment	Single Connection	8-Thread Parallel	32-Threads
LAN (Local)	~117 Mbps	~137 Mbps	~114 Mbps
WAN (Sweden - High Latency)	~7 Mbps	~42 Mbps	~71.44 Mbps
WAN (Newark, NJ - Low Latency)	~35 Mbps	~72 Mbps	~111.68 Mbps
Speedtest.net (Ultra-Low Latency)	N/A	93.19 Mbps	



Project is done in

gRPC + Spring Boot – Quick Intro

gRPC = communication framework

- Defines services and RPC methods (like remote function calls)
- Don’t have to deal with different language-specific libraries.
- Implements connections using HTTP 2.0 for us.

Spring Boot

- Java framework for building backend applications.
- Reduces boilerplate: handles app startup, object creation, and wiring services


Socials!
we made for two


CORNELL
CORNELL

cornelldata.science/recruitment

Website: cornelldata.science

Email: cornelldatascience@gmail.com

Instagram: @cornelldatascience

GitHub: github.com/CornellDataScience

Showcase: Friday, May 8th 12-4pm
@ Hollister B14

Instagram

Evaluating LLM Diagnosis with DAGs

Katie Xiao

Share

I've been feeling chills all day am i gonna be ok

You'll likely be okay, but here are some possible causes:

- Early-stage Influenza
- Common Cold
- COVID-19
- Fever starting or breaking
- Mild dehydration
- Not eating enough / low blood sugar
- Fatigue or poor sleep
- Anxiety / adrenaline response
- Hormonal fluctuations
- Being physically cold
- Strep Throat
- Urinary Tract Infection
- Mononucleosis
- Food Poisoning
- Lyme Disease
- Thyroid issues (e.g., Hypothyroidism)
- Mild anemia
- Medication side effects

Ask anything

Instant

Share

I just started violently coughing

You'll still likely be okay, but chills + sudden coughing usually points to a respiratory issue. Here are possible causes:

- Pneumonia
- Acute Bronchitis
- Respiratory Syncytial Virus
- Whooping Cough
- Asthma flare
- Airway irritation (dust, cold air, dryness)
- Post-nasal drip triggering coughing fits
- Sudden airway constriction
- Pulmonary Embolism
- Tuberculosis
- Sepsis

If you have shortness of breath, chest pain, feel faint, or can't catch your breath between coughing, get urgent care.

Ask anything

Instant

How accurately do LLMs update their diagnoses as new patient information is revealed?

We need to create some form of “reference” model

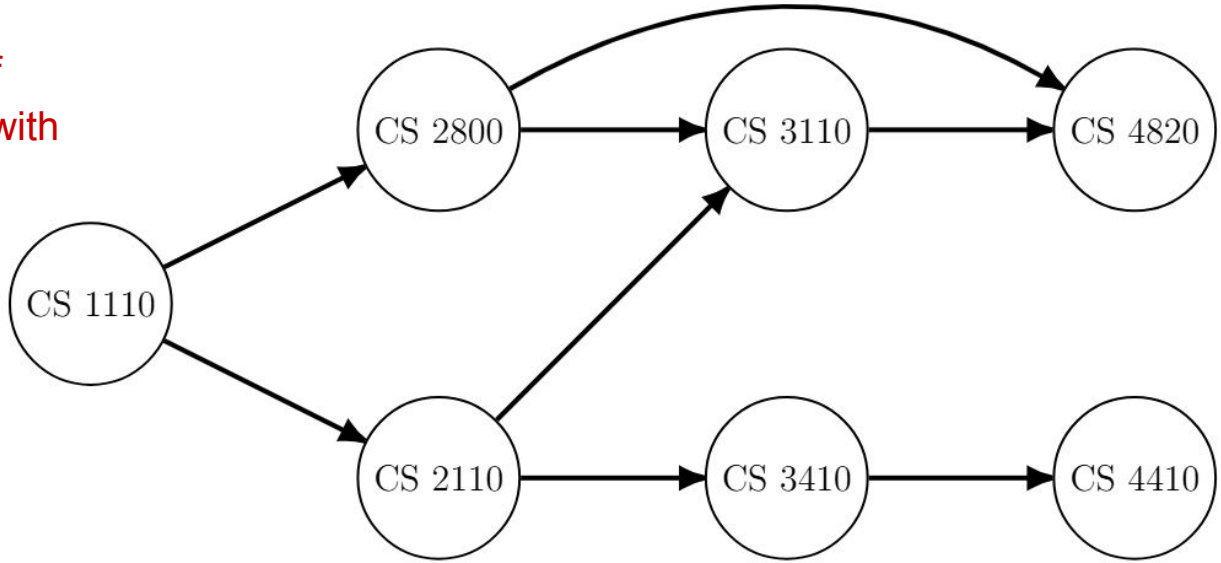
How **accurately** do LLMs update their diagnoses as new patient information is revealed?

Directed Acyclic Graph (DAG)

A directed graph with no cycles

Many things can be modelled with DAGs!

(Exercise 22.7: A graph of core Cornell CS courses with edges representing prerequisites)

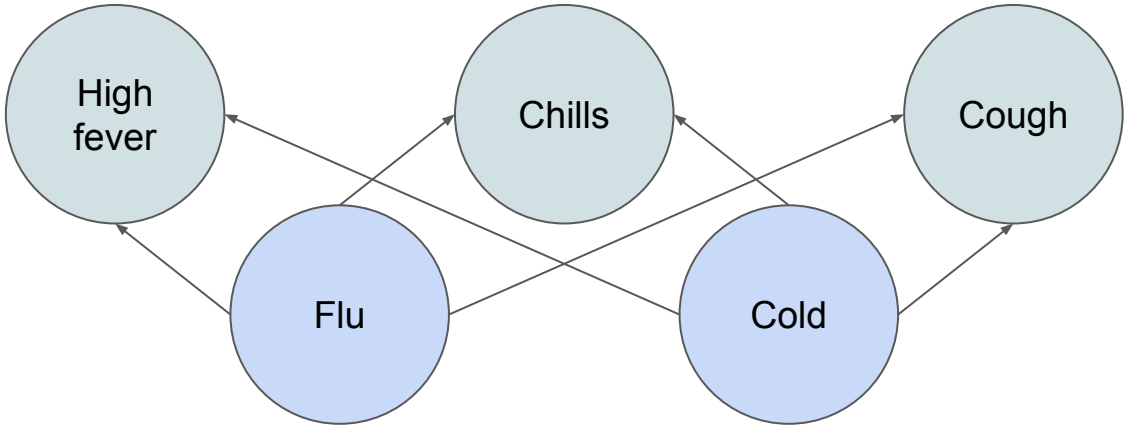


DAGs for Diagnostic Reasoning

DAGs are often used to model **causal relationships**.

In the context of diagnostic reasoning:

- Nodes = Diseases and findings (e.g. symptoms, test results, etc.)
- Edges = [Disease] can be caused by [finding]



DAGs for Diagnostic Reasoning

Problem 1: The doctor is **never 100% certain** that a patient has a certain disease

Solution: Model the diagnosis as a **conditional probability:**

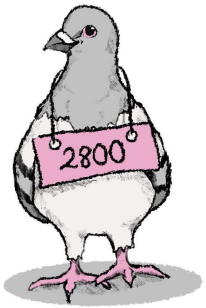
$$Pr(\text{disease} \mid \text{symptom})$$

(What is the probability that the patient has [disease] **given** that they have [symptom]?)

DAGs for Diagnostic Reasoning

Problem 2: This conditional probability is difficult to estimate directly from data.

$$Pr(\text{disease} \mid \text{symptom})$$



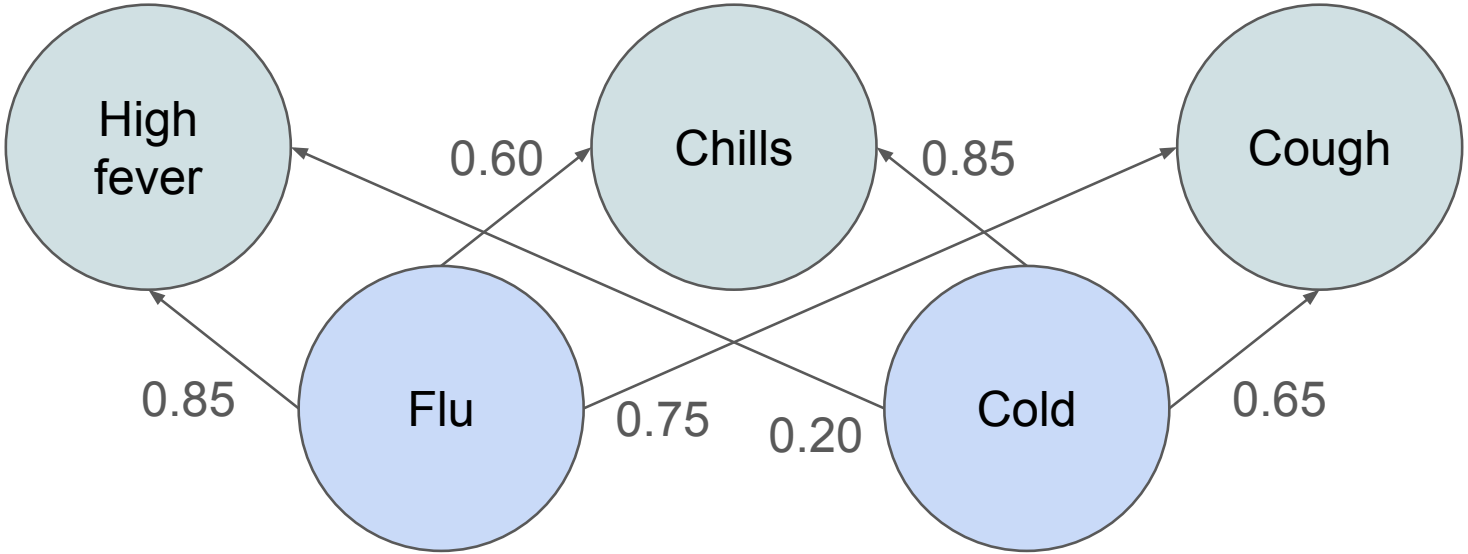
Solution: Use Bayes' Rule!

$$Pr(\text{disease} \mid \text{symptom}) = \frac{Pr(\text{disease}) \cdot Pr(\text{symptom} \mid \text{disease})}{Pr(\text{symptom})}$$

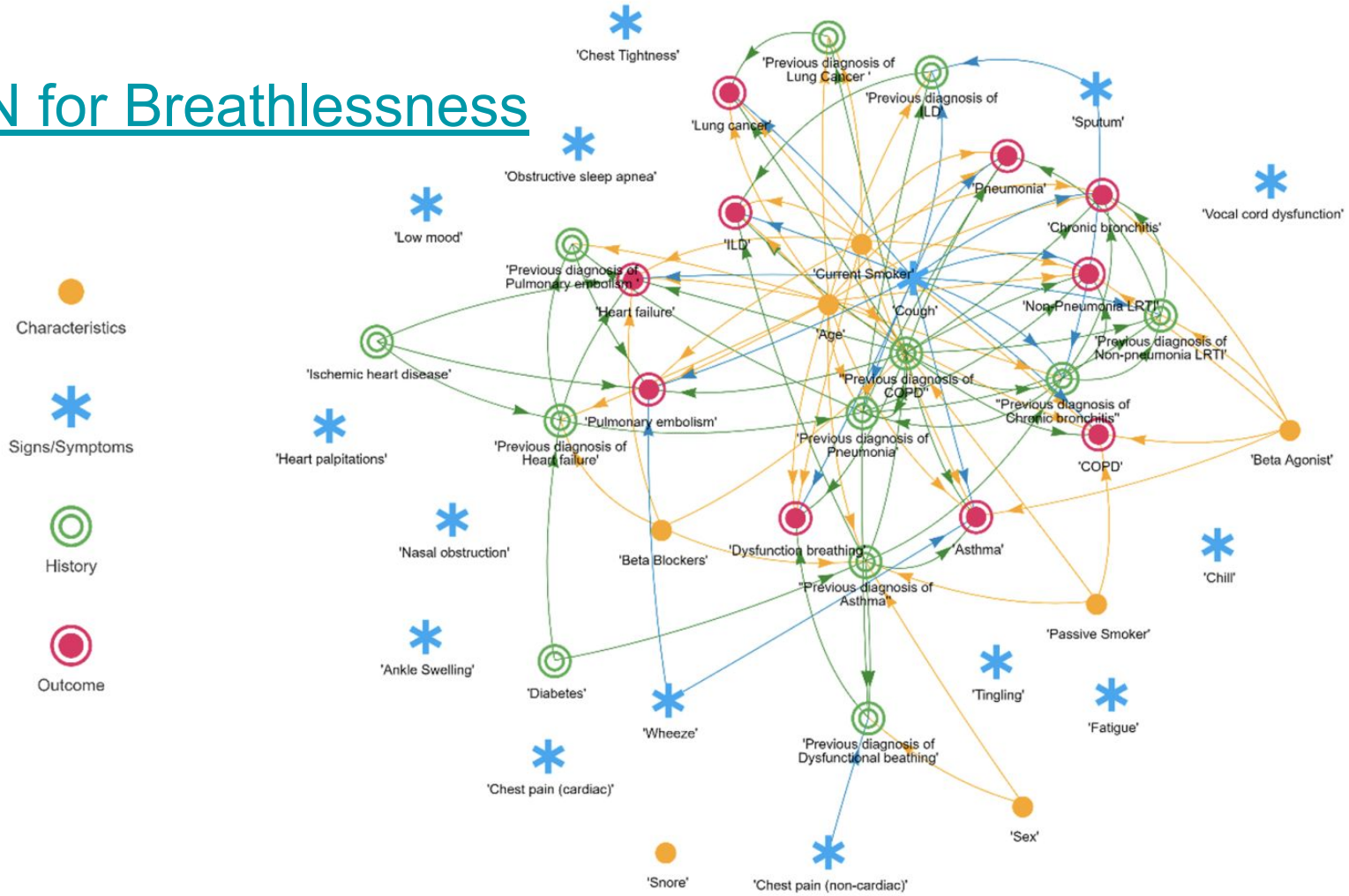
(This conditional probability can be more easily estimated using medical data.)

Bayesian Networks

Bayesian Network (BN): A DAG where edges represent conditional probabilities between variables



A BN for Breathlessness



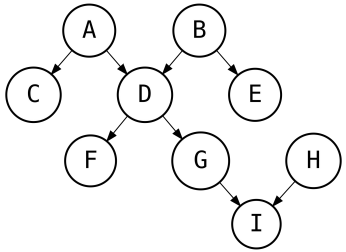
Bayesian Networks and Diagnostic Reasoning

A key advantage of using BNs as a “ground truth” model for evaluating LLMs is that they support **step-by-step updates**, similar to how a doctor updates their diagnosis as they receive new patient information!

Bayesian Networks and Diagnostic Reasoning

Questions I am researching:

- How should we preprocess medical data to construct a BN that accurately reflects real-world relationships between diagnoses and symptoms?”
- Using a BN as a reference, how can we design evaluation tasks that measure how well LLMs perform diagnostic reasoning?



CUAir - Autopilot

Alex & Emily

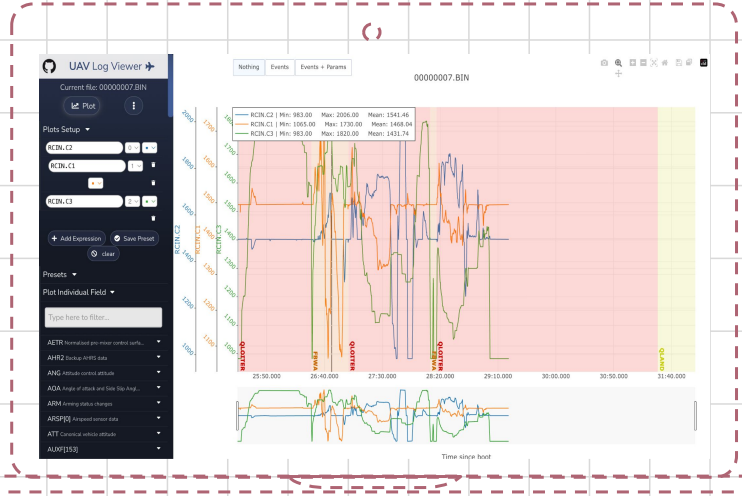
CUAir Project Team

Cornell Unmanned Air Systems:

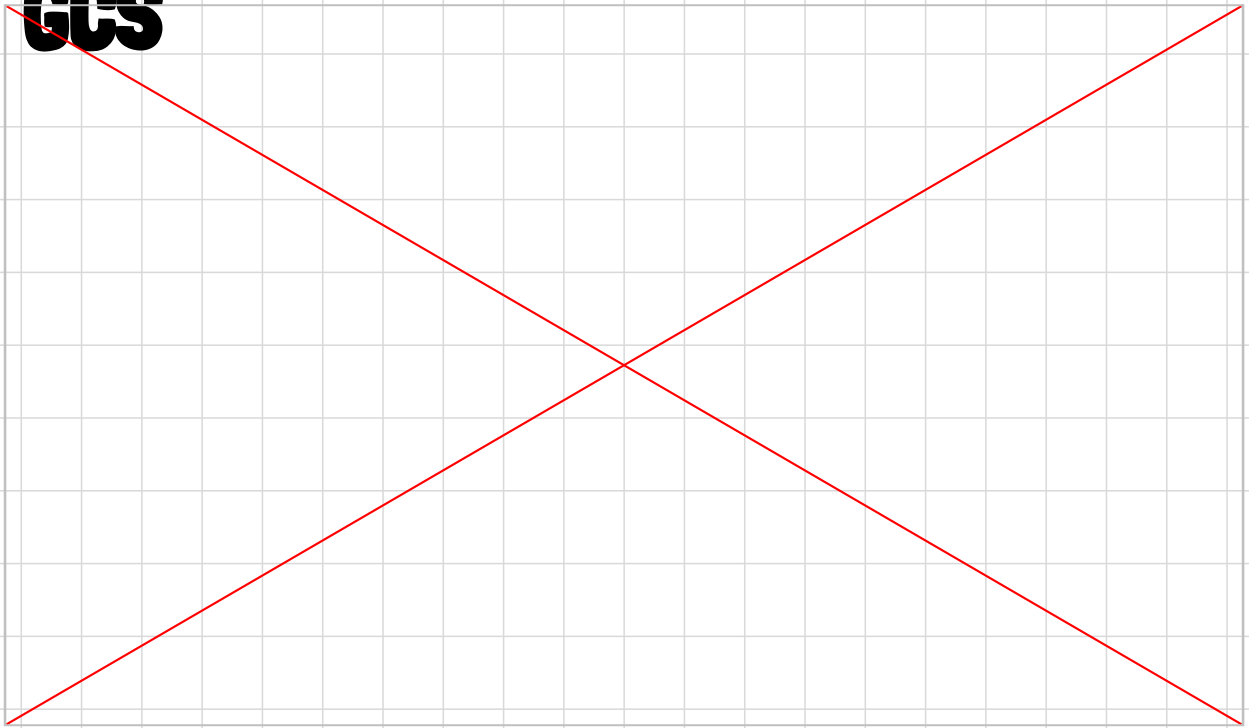
We design, build, code, and fly an autonomous quadplane for the SUAS competition every year

Autopilot Subteam

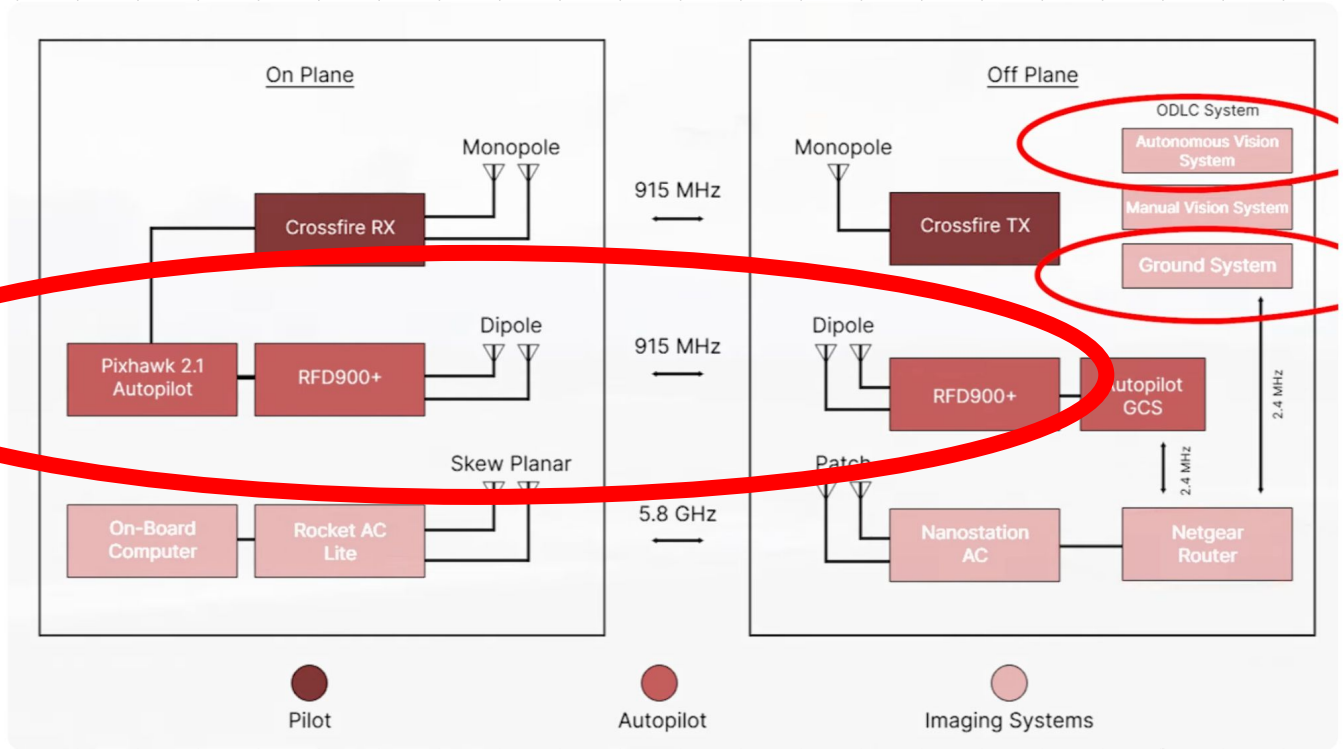
- Ground Control Station (GCS)
- Testing and calibration
- Intersection of software and hardware

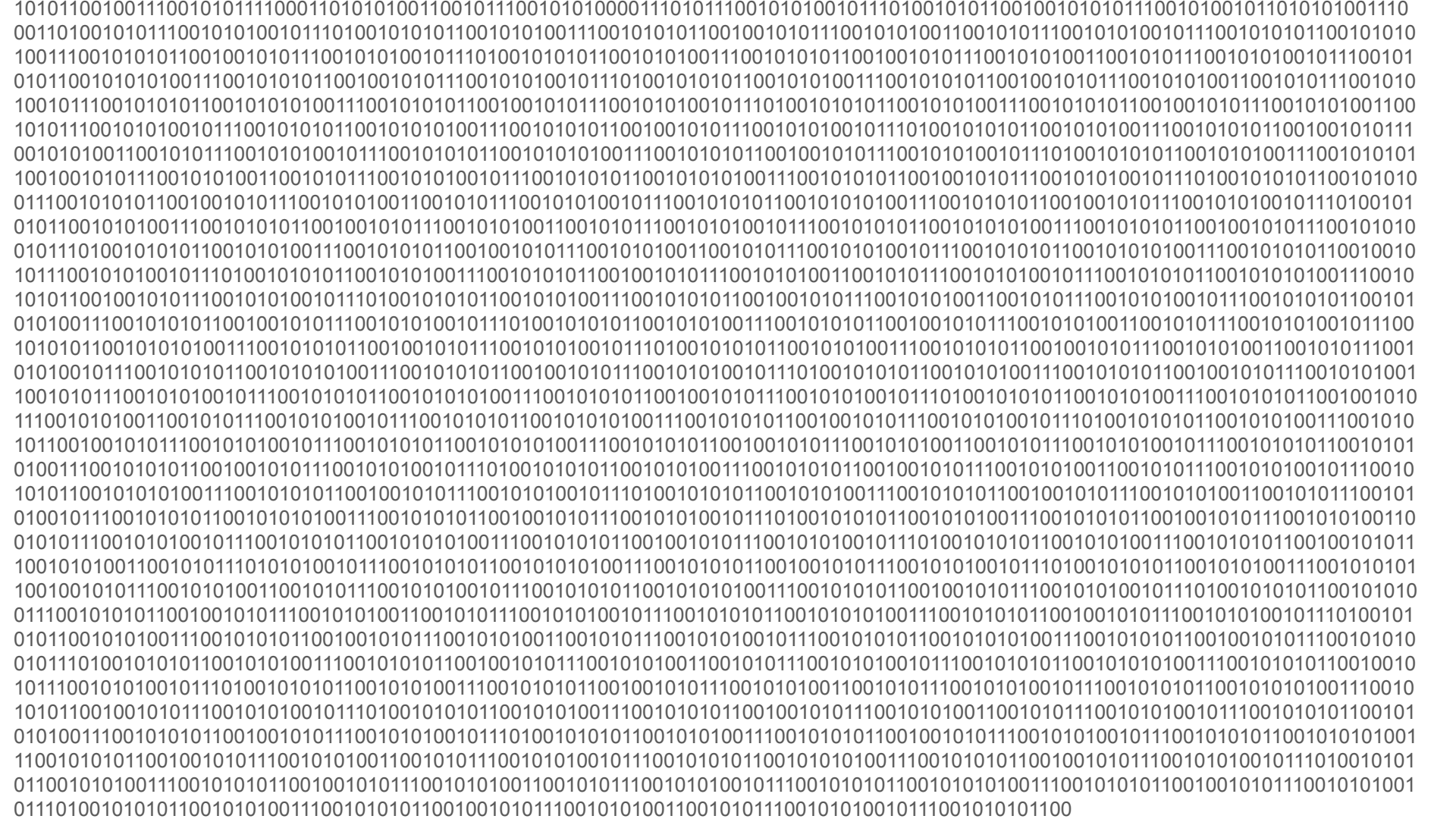


Demo of GCS

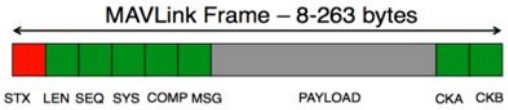


How do we "talk" with our plane?





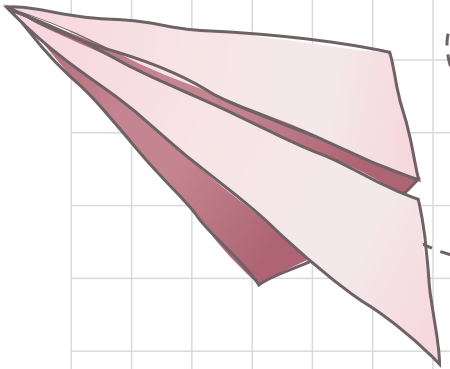
Transmitting serialized data is hard!



Byte Index	Content	Value	Explanation
0	Packet start sign	v1.0: 0xFE (v0.9: 0x55)	Indicates the start of a new packet.
1	Payload length	0 - 255	Indicates length of the following payload.
2	Packet sequence	0 - 255	Each component counts up his send sequence. Allows to detect packet loss
3	System ID	1 - 255	ID of the SENDING system. Allows to differentiate different MAVs on the same network.
4	Component ID	0 - 255	ID of the SENDING component. Allows to differentiate different components of the same system, e.g. the IMU and the autopilot.
5	Message ID	0 - 255	ID of the message - the id defines what the payload "means" and how it should be correctly decoded.
6 to (n+6)	Data	(0 - 255) bytes	Data of the message, depends on the message id.
(n+7) to (n+8)	Checksum (low byte, high byte)	ITU X.25/SAE AS-4 hash, **excluding packet start sign, so bytes 1..(n+6)** Note: The checksum also includes MAVLINK_CRC_EXTRA (Number computed from message fields. Protects the packet from decoding a different version of the same packet but with different variables).	

MAVProxy

Enter.....

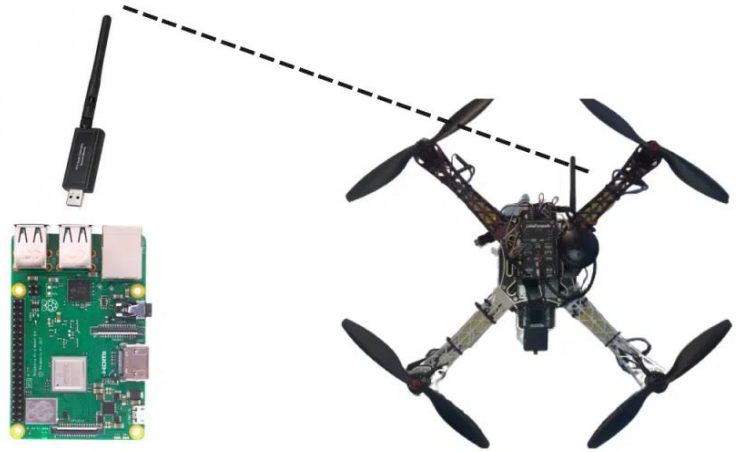


MAVLINK

MICRO AIR VEHICLE COMMUNICATION PROTOCOL

What is MAVProxy?

- Lightweight interface we use to outsource serialization
- Provides convenient methods to encode information into messages without having to remember lots of numbers
  - Refer to different waypoints or mission items as constants instead of by number
  - EX: MAV_CMD_DO_VTOL_TRANSITION is command 3000!
- Modules all instances of MPModule class -> allows for polymorphic benefits!



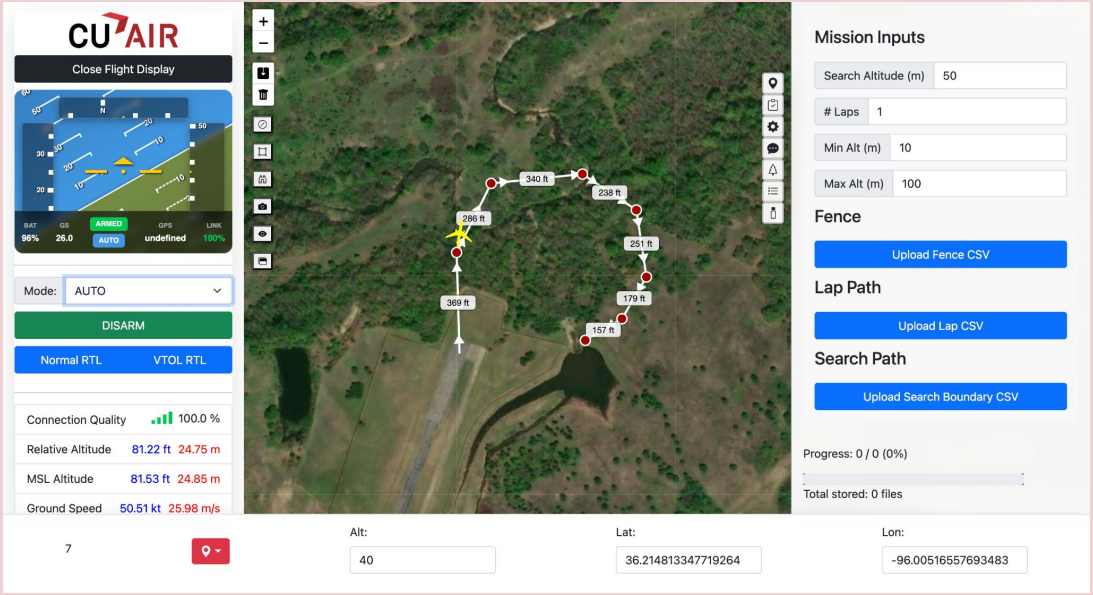
Example: creating waypoints

```
def _create_wp(self, wp):
    special = [ mavutil.mavlink.MAV_CMD_DO_SET_SERVO, mavutil.mavlink.MAV_CMD_NAV_LOITER_TIME,
                mavutil.mavlink.MAV_CMD_DO_VTOL_TRANSITION, mavutil.mavlink.MAV_CMD_NAV_VTOL_LAND,
                mavutil.mavlink.MAV_CMD_NAV_VTOL_TAKEOFF, mavutil.mavlink.MAV_CMD_NAV_RETURN_TO_LAUNCH
                ]
    try:
        if wp['command'] not in special: # NAV WAYPOINT
            p = mavutil.mavlink.MAVLink_mission_item_message(
                self.wploader.target_system,
                self.wploader.target_component,
                0,
                mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT,
                wp['command'],
                wp['current'],
                1, 0, 0, 0, 0,
                wp['lat'], wp['lon'], wp['alt'])
        elif wp['command'] == mavutil.mavlink.MAV_CMD_DO_SET_SERVO: # SERVO WAYPOINT
```

Poll Everywhere

PollEv.com/javabear text **javabear** to 22333

Which is the best data structure for representing a flight path?



Stack **(A)**

BinaryTree **(B)**

HashMap **(C)**

ArrayList **(D)**

CUAUV - Infra

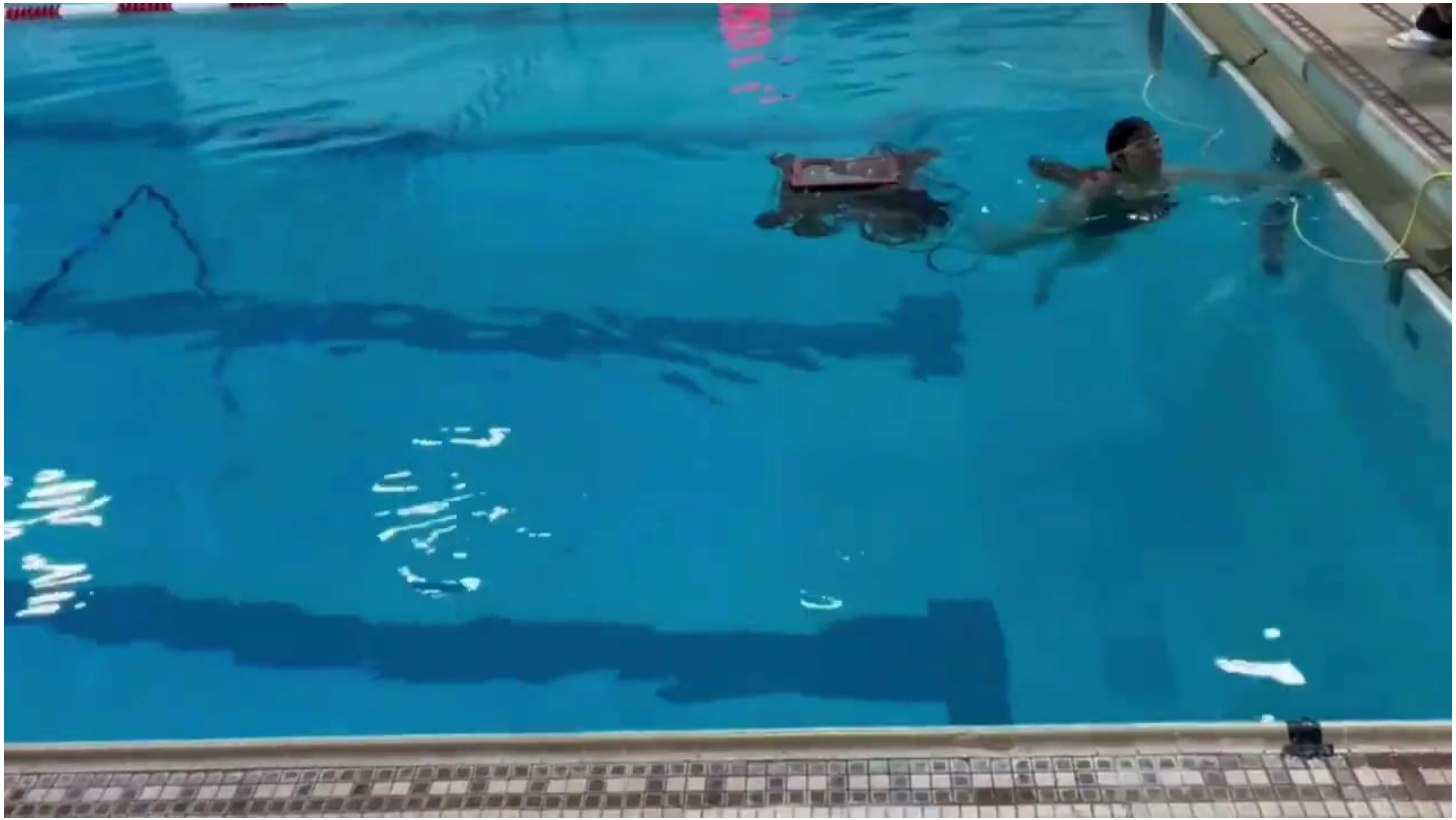
Kyle Du

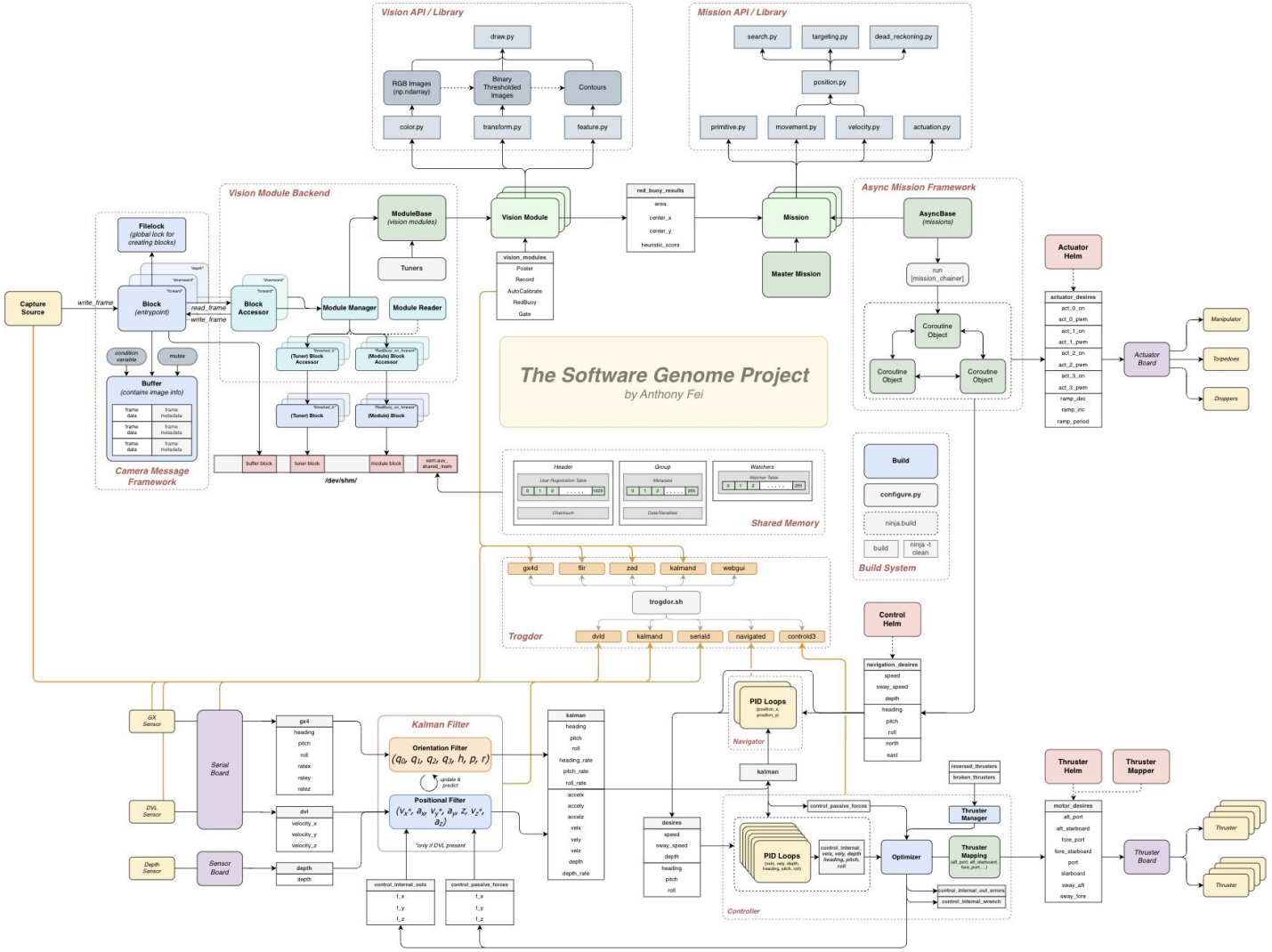
CU AUV

Cornell University
Autonomous Underwater Vehicles
Project Team

We build submarines, equip them with a bunch of sensors, and write code to make them perform autonomous tasks!

Some footage courtesy of business



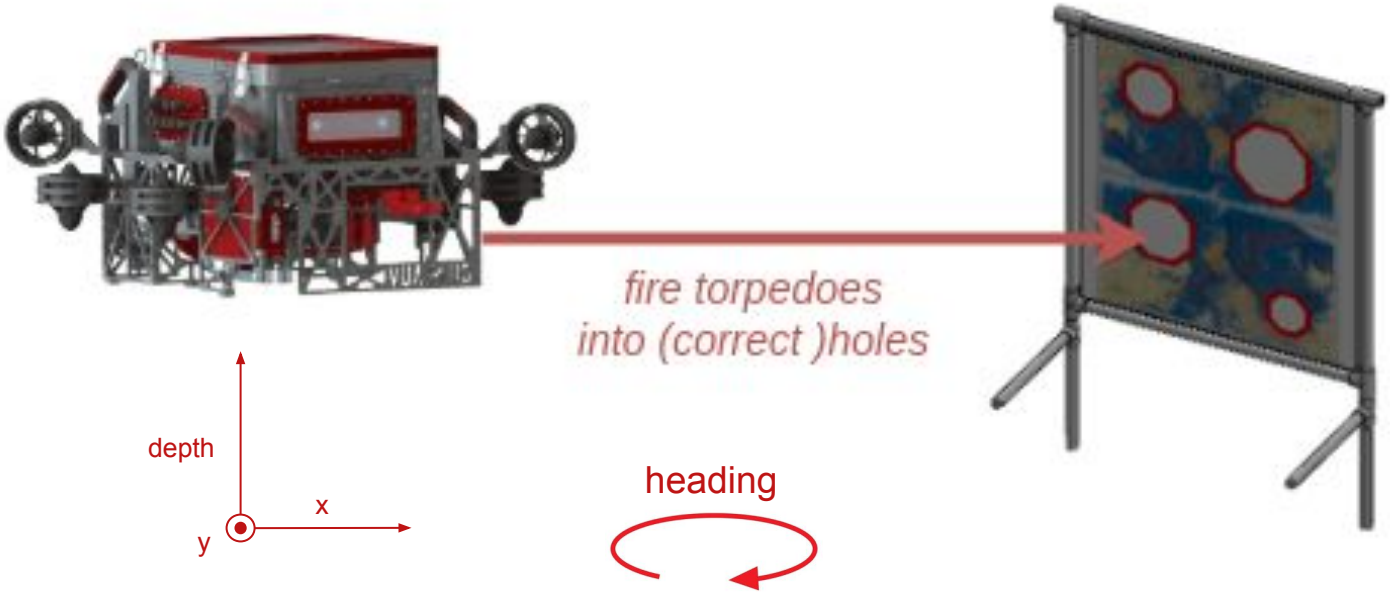


High-level overview of our software stack.

Lots of stuff going on. How do we do everything all at once?

Concurrency!

Mission System: Concurrent Steps



Python Concurrency: `Asyncio`

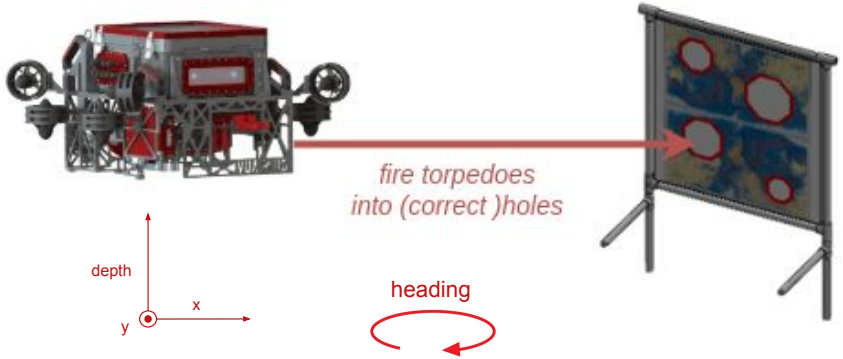
Python's `asyncio.create_task()` is similar to starting a "lightweight" Java thread.

```
Thread t = new Thread(task);
t.start();

task = asyncio.create_task(coroutine)
```

Python's `await` is similar to Java `join()` but for coroutines.

```
thread.join();

await coroutine
```

Disclaimer: Asyncio and Java threads are actually very different in how they achieve concurrency. But for the purpose of this demo, they are similar in purpose.




```
Thread positionXThread = new Thread(() -> moveXToTarget());
Thread positionYThread = new Thread(() -> moveYToTarget());
Thread headingThread = new Thread(() -> alignHeadingToBoard());
Thread depthThread = new Thread(() -> setTorpedoDepth());

positionXThread.start();
positionYThread.start();
headingThread.start();
depthThread.start();

positionXThread.join();
positionYThread.join();
headingThread.join();
depthThread.join();

Thread holdInPlaceThread = new Thread(() -> holdPosition());
Thread fireTorpedoesThread = new Thread(() -> fireTorpedoes());

holdInPlaceThread.start();
fireTorpedoesThread.start();

holdInPlaceThread.join();
fireTorpedoesThread.join();
```

Vision System: Synchronization

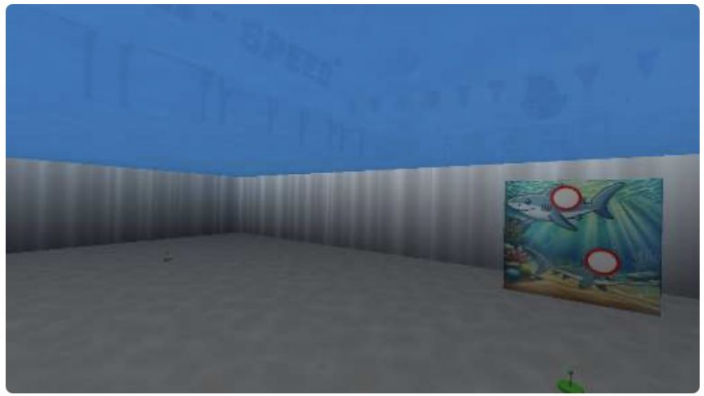
← Back to modules

YoloTorpedoes-on-zed

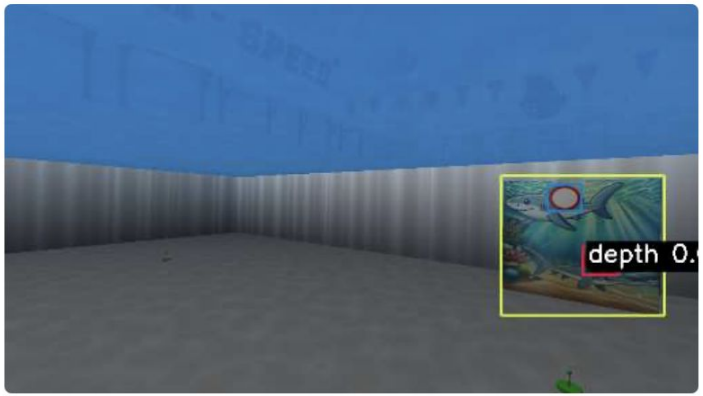
What the submarine sees

Images

original image 6.9 FPS



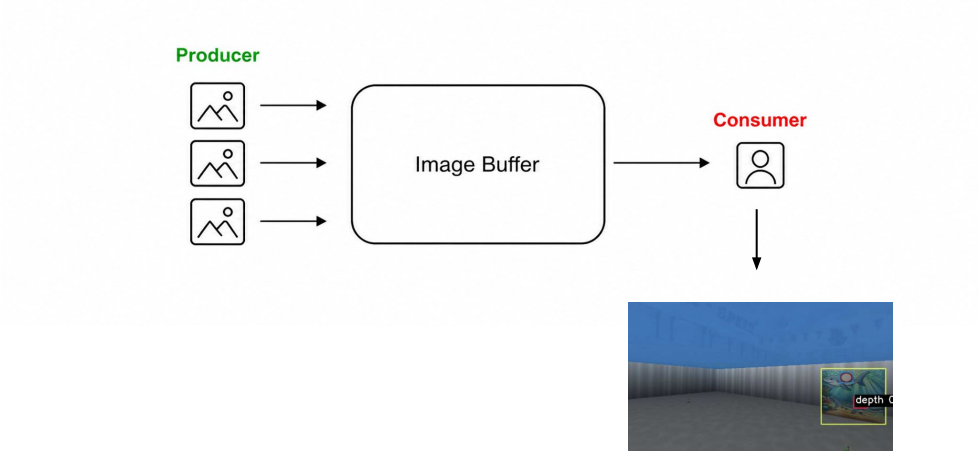
torpedoes handler 7 FPS



Python Synchronization: `threading.Lock`

Our vision modules use multiple threads (not `asyncio`-based, more like true Java threads).

- Image buffer to queue incoming frames
- "Producer" threads
  - Feed image frames into the buffer
- "Consumer" threads
  - Process the frames and clear them (e.g. for YOLO-detection)



Use Python's `threading.Lock` to make the image buffer thread-safe.

```
self._post_queue = OrderedDict()
self._post_lock = threading.Lock()
```

Producer access

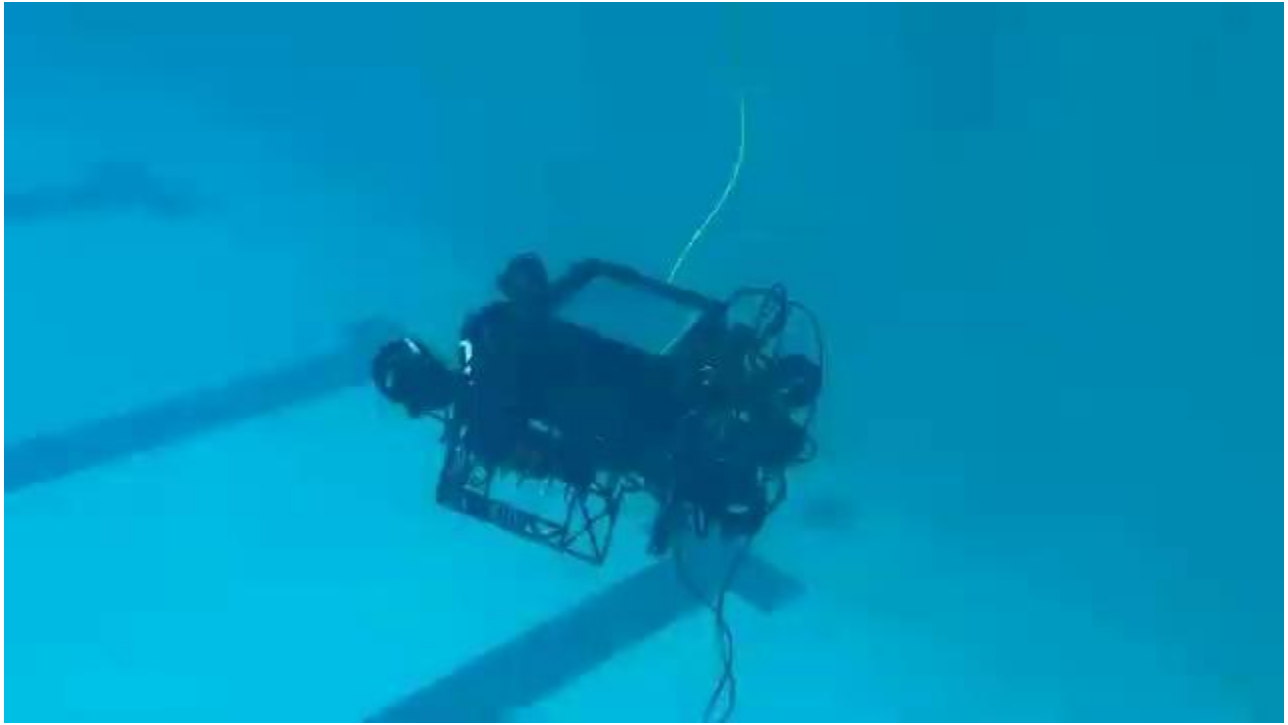
```
with self._post_lock:
    self._post_queue[name] = image
```

Consumer access

```
with self._post_lock:
    items = list(self._post_queue.items())
    color_spaces = dict(getattr(self, "_post_color_spaces", {}))
    self._post_queue.clear()
    if hasattr(self, "_post_color_spaces"):
        self._post_color_spaces.clear()
```

Locks help prevent one thread from writing to the buffer while another is reading from it; that could cause a race condition

barrel roll whee

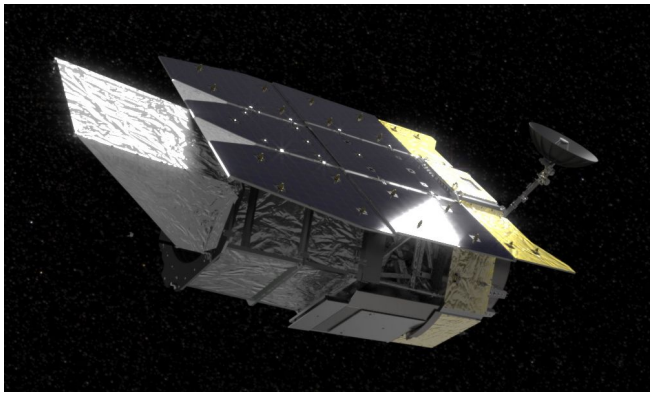


Astro and CS

Abrar Amin

Astronomy and Computer Science?

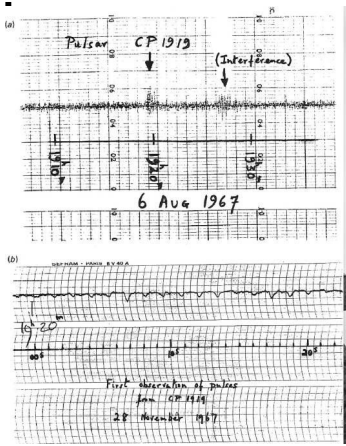
- Astronomy is a field which encompasses a broad range of subfields of pretty much anything in the sky/universe.
- Until the early 1990s, most astronomers actually observed through telescopes to collect data on things in the sky they are interested in (galaxies, stars, planets, and more!)



Astronomy and Computer Science?

- Just like the most of the world, the field of Astronomy has been shaped with the advent of digital technology!
- Telescopes provide data automatically/digitally, so astronomers can do all of their research all remotely.
  - But this also created a new issue…

This is Jocelyn Bell Burnell (the discoverer of pulsars!) who used an analog telescope that printed data onto a piece of paper



The Scale of “Big Data”

- Vera C. Rubin Observatory (LSST): Captures 3.2-gigapixel images every 40 seconds, totaling **10 terabytes** nightly and **60 million billion bytes (60 petabytes) over a decade.**
  - There are all kinds of scientists (exoplanet, supernovae, galaxy, and other fields) who will use this information! It is **important** for all this data to be organized and to be accessible universally!

Poll Everywhere

PollEv.com/javabear text **javabear** to 22333

Let's assume the Vera C. Rubin Observatory can detect **1 billion objects** in the sky per night. Let us also assume there are the following algorithms we run on the dataset for one night:

AstroSearch: O(log N)

StarFind: O(N^2)

Approximately how many steps would each algorithm take for N = 1,000,000,000?

log N ≈ 30 steps, N^2 ≈ 10^(18) steps **(A)**

log N ≈ 1,000 steps, N^2 ≈ 10^9 steps **(B)**

log N ≈ 10^9 steps, N^2 ≈ 30 steps **(C)**

log N ≈ 67 steps, N^2 ≈ 68 steps **(D)**

Poll Everywhere

PollEv.com/javabear text **javabear** to 22333

Let's assume the Vera C. Rubin Observatory can detect **1 billion objects** in the sky per night. Let us also assume there are the following algorithms we run on the dataset for one night:

AstroSearch: O(log N)

StarFind: O(N^2)

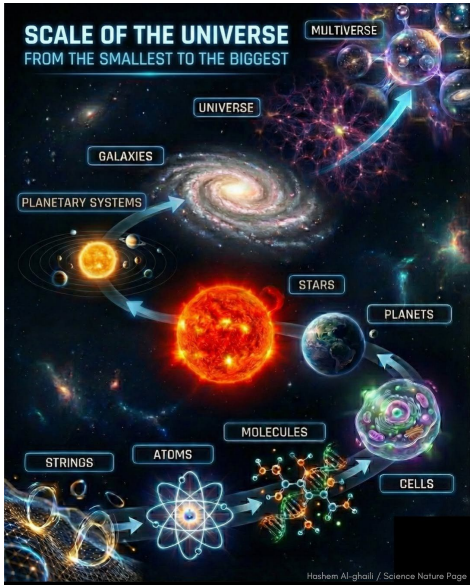
Approximately how many steps would each algorithm take for N = 1,000,000,000?

- **log N ≈ 30 steps, N^2 ≈ 10^(18) steps** **(A)**
- **log N ≈ 1,000 steps, N^2 ≈ 10^9 steps** **(B)**
- **log N ≈ 10^9 steps, N^2 ≈ 30 steps** **(C)**
- **log N ≈ 67 steps, N^2 ≈ 68 steps** **(D)**

Cosmic Hierarchy: More or Less

Planets → Star System (Solar System) → **Star Cluster** → Galaxy → Galaxy Group → Galaxy Cluster → ... → Universe

- Each level is its own cool and interesting field, but we will focus on star clusters today.



The universe is basically just a graph…

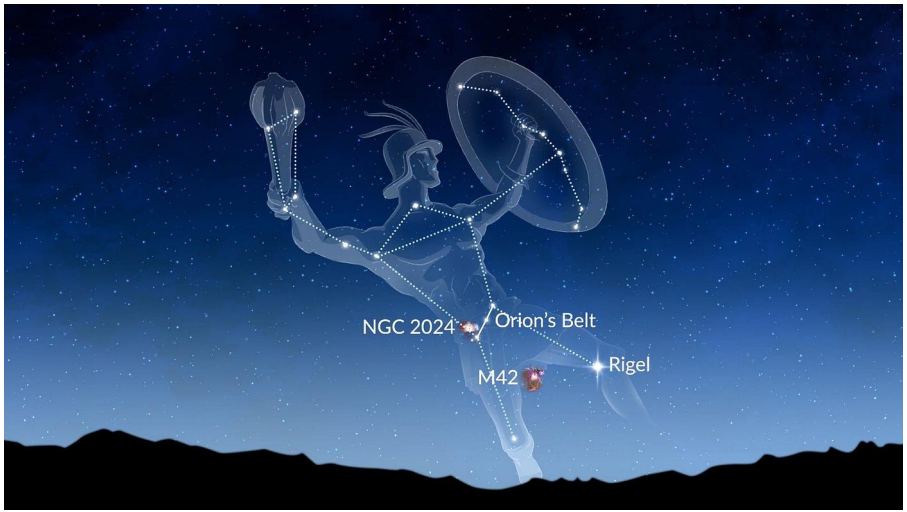
A star cluster is a group of stars that formed together from the **same molecular gas cloud**, **sharing a common age** and **chemical composition**, and are held together by gravity

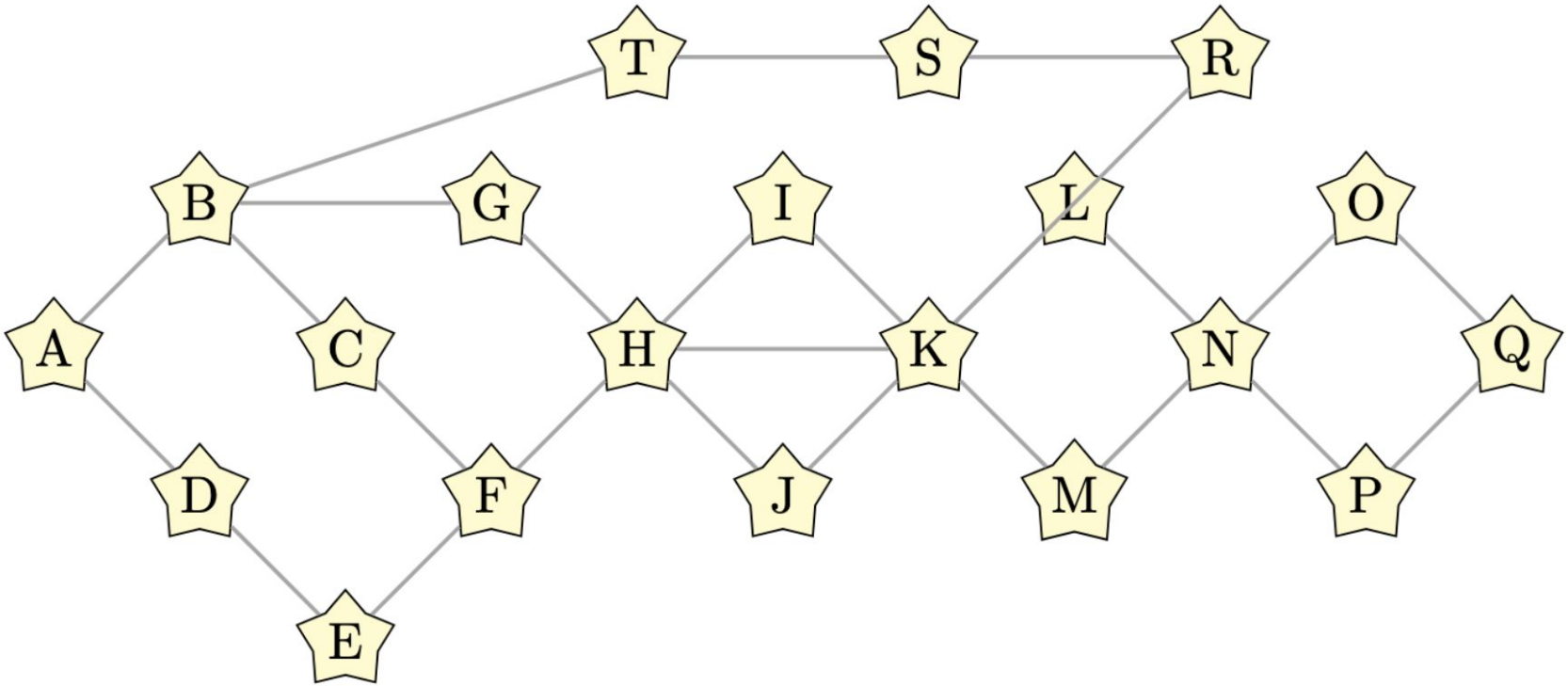
- Figuring out if certain stars are in the same star cluster can tell scientists a lot about a particular star’s nature and how they formed…
- If we can represent Pac-Mann with graphs, why not stars?

The universe is basically just a graph…

- In terms of CS 2110, we can liken these clusters in terms of a graphs with **stars as nodes** and **edges denoting clusters.**

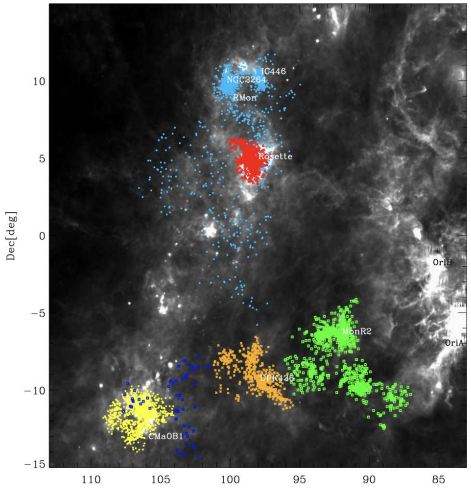
- Stars are really nodes, and stars within the same cluster should have an edge connecting each other. (Constellations **ARE NOT** star clusters, but they are an example of the graph analogy)




T
S
R
B
G
I
L
O
A
C
H
K
N
Q
D
F
J
M
P
E

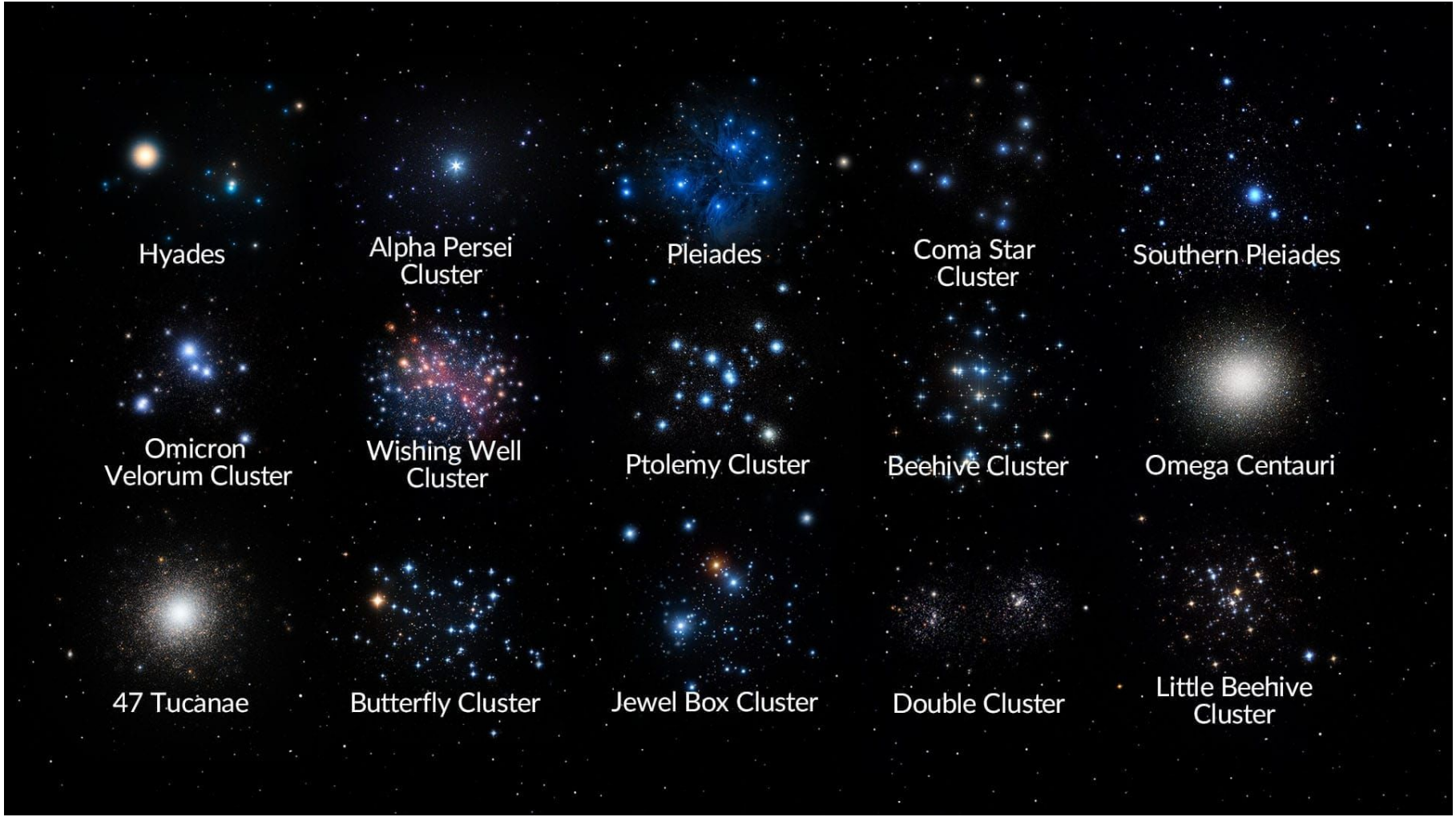
The universe is basically just a graph…

- Scientists have used clustering algorithms (take CS 3780/ASTRO 4523 if interested!) to help identify these star clusters.
- The same concepts of graph traversals also apply here!
  - Within a cluster, we identify any stars closest neighbor!
  - We can start asking more questions about these clusters:
    - How do stars inside the cluster interact locally?
    - What kinds of structures form within these clusters?



The universe is basically just a graph…

- Edges within a cluster can represent a multitude of things:
    - Gravitational interaction strength between stars, where edge weights reflect the magnitude of gravitational influence.
    - Spatial proximity, where edges connect stars that are close in three-dimensional position space.
    - Phase-space similarity, where edges connect stars that are close in both position and velocity, indicating a likely common dynamical origin (originating from the same molecular cloud)


Hyades
Alpha Persei Cluster
Pleiades
Coma Star Cluster
Southern Pleiades
Omicron Velorum Cluster
Wishing Well Cluster
Ptolemy Cluster
Beehive Cluster
Omega Centauri
47 Tucanae
Butterfly Cluster
Jewel Box Cluster
Double Cluster
Little Beehive Cluster

Lecture 28: Real-World Software Engineering

CS 2110

May 5, 2026