Welcome

to

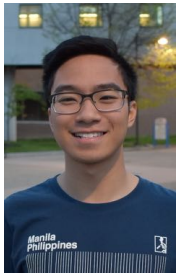
CSE 331

Let’s do some introductions

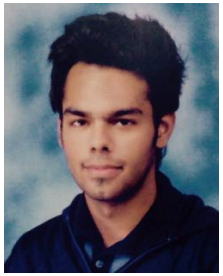
http://www.zazzle.com/warning_teaching_assistant_bag-149882665435161818

TAs first

Chinmayee



Hans



Sanchit

Priyanka

Elijah

Stephen

Nick

Sean

Rishi

Gitanjali

Supratik

Animesh

Tom

Veronica

Lectures will be videotaped

http://www.imdb.com/title/tt0363510/mediaviewer/rm2499681792

About Me

Atri Rudra

atri@buffalo.edu

Office: 319 Davis

Office hours: Mon 2-2:50pm; Wed, 3:00-3:50pm

OH starts today

Contact us all at

cse-331-staff@buffalo.edu

TAs will not respond to individual emails (except for re-grading requests)

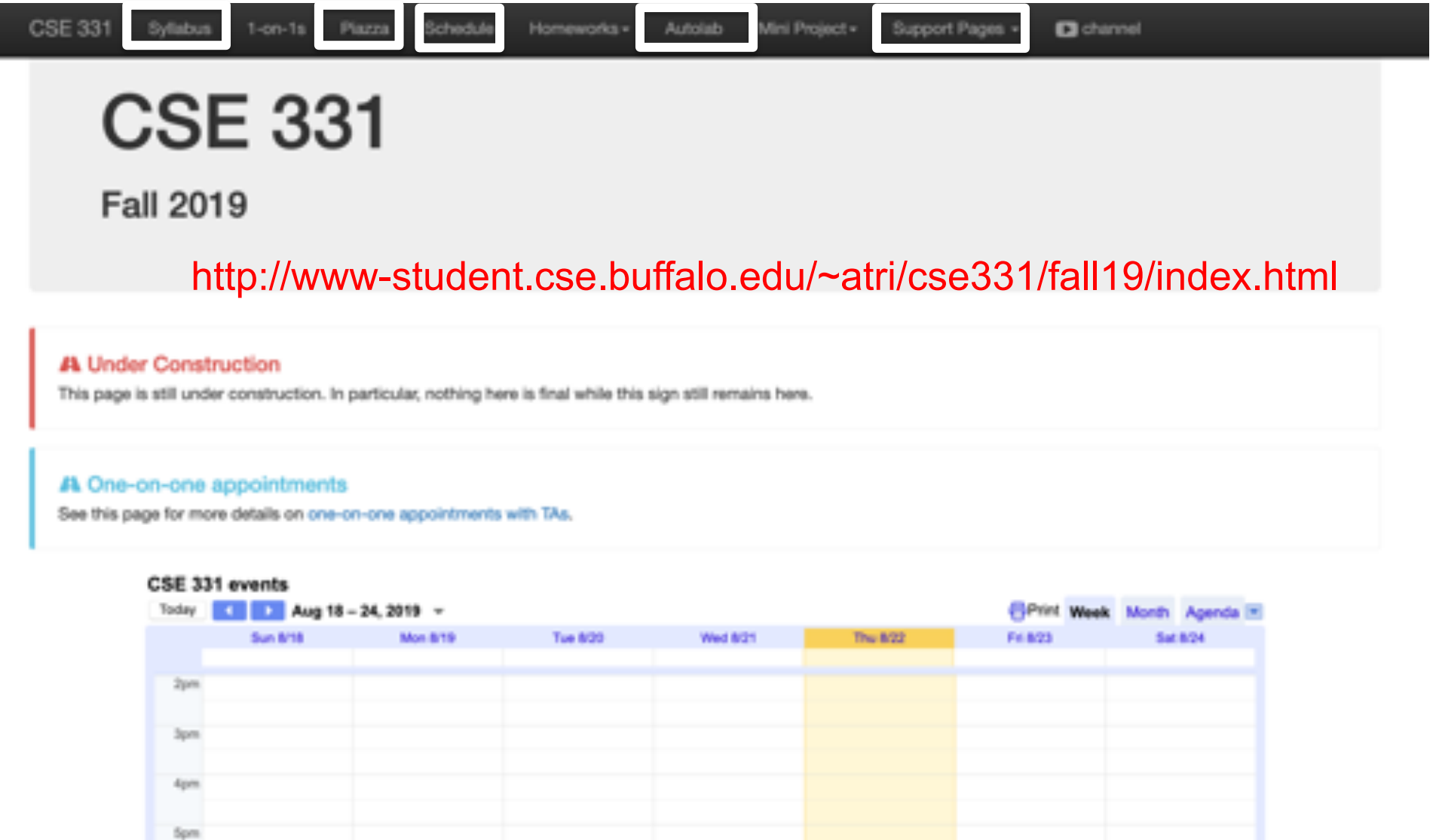
Handouts for today

Syllabus (online)

Homework Policy document (online)

Homework 0 (online)

One Stop Shop for the Course



Homework 0 (Optional)

CSE 331 Syllabus 1-on-1s Piazza Schedule Homeworks Autolab Mini Project Support Pages channel

HW 0

Allowed Sources

Homework Policies

Homework (

Due by **11:00am, Friday, August 30, 2019**.

Due: this Friday 11am

Make sure you follow all the homework policies.

All submissions should be done via Autolab

The support page for matrix vector multiplication could be very useful for this homework.

Submitting HW 0 is optional. However, we do encourage you to submit to get familiar with Autolab and to get some feedback.

What is a proof?

The goal of this question is to present a gentle start to proofs. In particular, the idea is to highlight a common mistake students make while writing proofs.

The Problem

Consider the following "proof":

- Brad Pitt has a beard:

Brad Pitt

- Every goat has a beard:

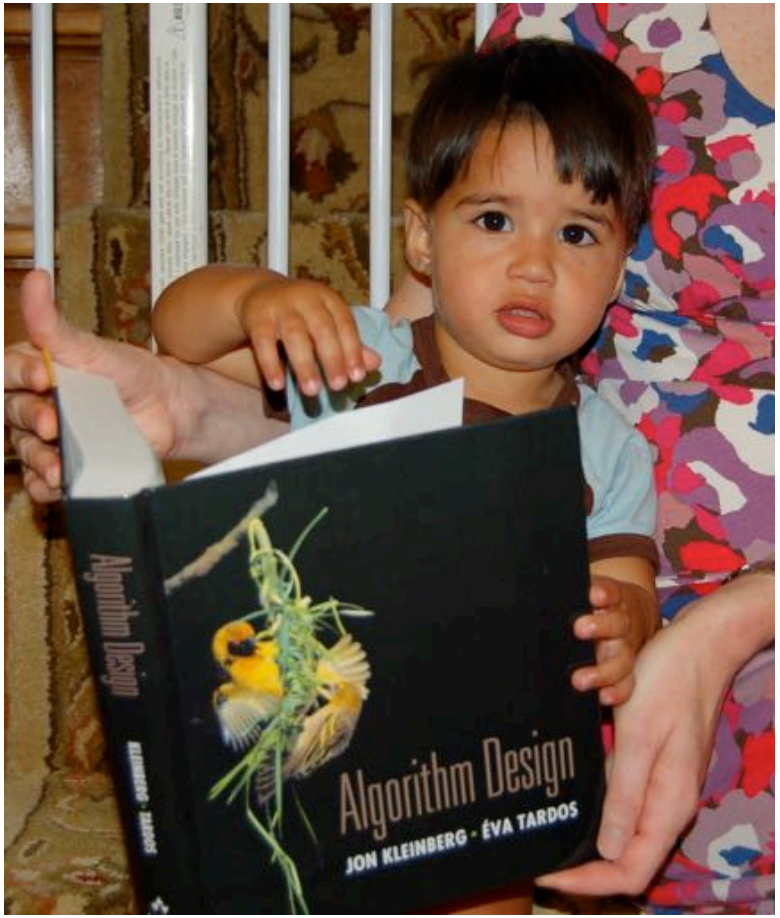
Three things to remember

WORK HARD!

DO NOT CHEAT!

READ CAREFULLY!

Wait.. What???



Make sure you follow submission instructions

Two most common ways of losing points

Make sure you read problem statements carefully

Academic Dishonesty

All your submissions must be your own work

Penalty:

Minimum: An **grade reduction in course**

Possible: **F** (or higher penalty) if warranted

YOUR responsibility to know what is cheating, plagiarism etc.

If not sure, come talk to me

Excuses like "I have a job," "This was OK earlier/in my country," "This course is hard," etc. **WON'T WORK**

I DO NOT HAVE ANY PATIENCE WITH ANY CHEATING :

YOU WILL GET A GRADE REDUCTION IN THE COURSE

FOR YOUR FIRST MISTAKE

Read the syllabus CAREFULLY!

Syllabus Quiz

No graded material will be handed back till you pass the syllabus quiz!

Options

View handin history

View writeup

Download handout

Due: **December 12th 2019, 4:05 pm**

Last day to handin: **December 12th 2019, 6:05 pm**

Academic Integrity

Question 1: Sharing my answers to this syllabus quiz with other 331 students

- ○ Is OK if I do it to help out a friend
- ○ It does not matter since there is no grade attached with it
- ○ Is an academic integrity violation and should not be done
- ○ Is an academic integrity violation but I can take the chance

More information on the quiz

CSE 331 Syllabus

Algorithms and Complexity

Fall 2019

Time and location: **Mondays, Wednesdays** and **Fridays, 1:00-1:50pm**, **Norton 112**.

Under Construction

This page is still under construction. In particular, nothing here is final while this sign still remains here.

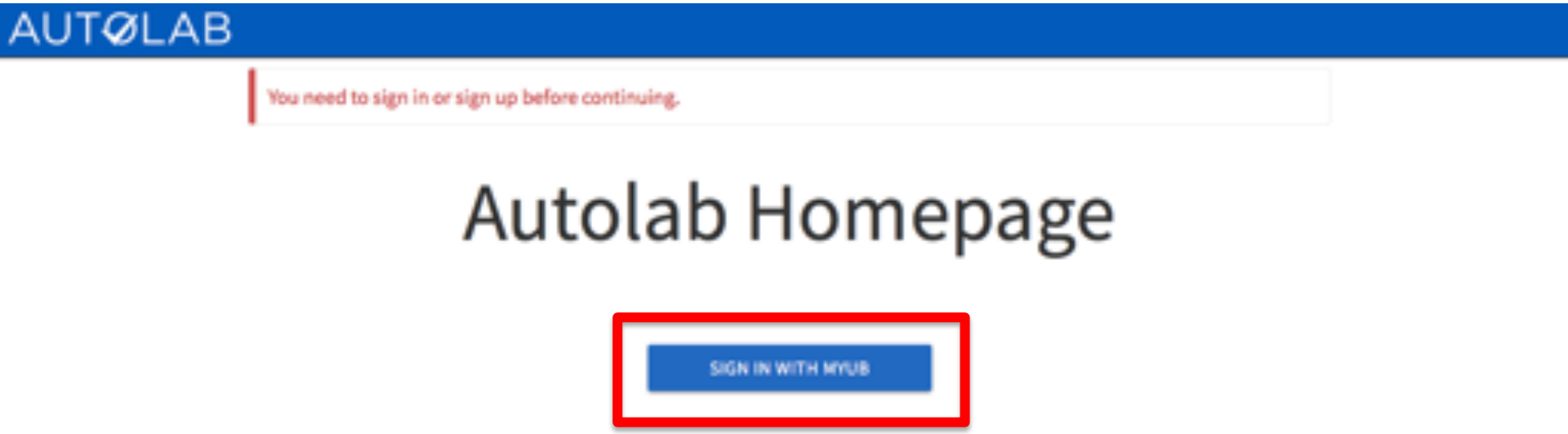
Please note

It is **your responsibility** to make sure you read and understand the contents of this syllabus. If you have any questions, please contact the instructor.

Acknowledgment

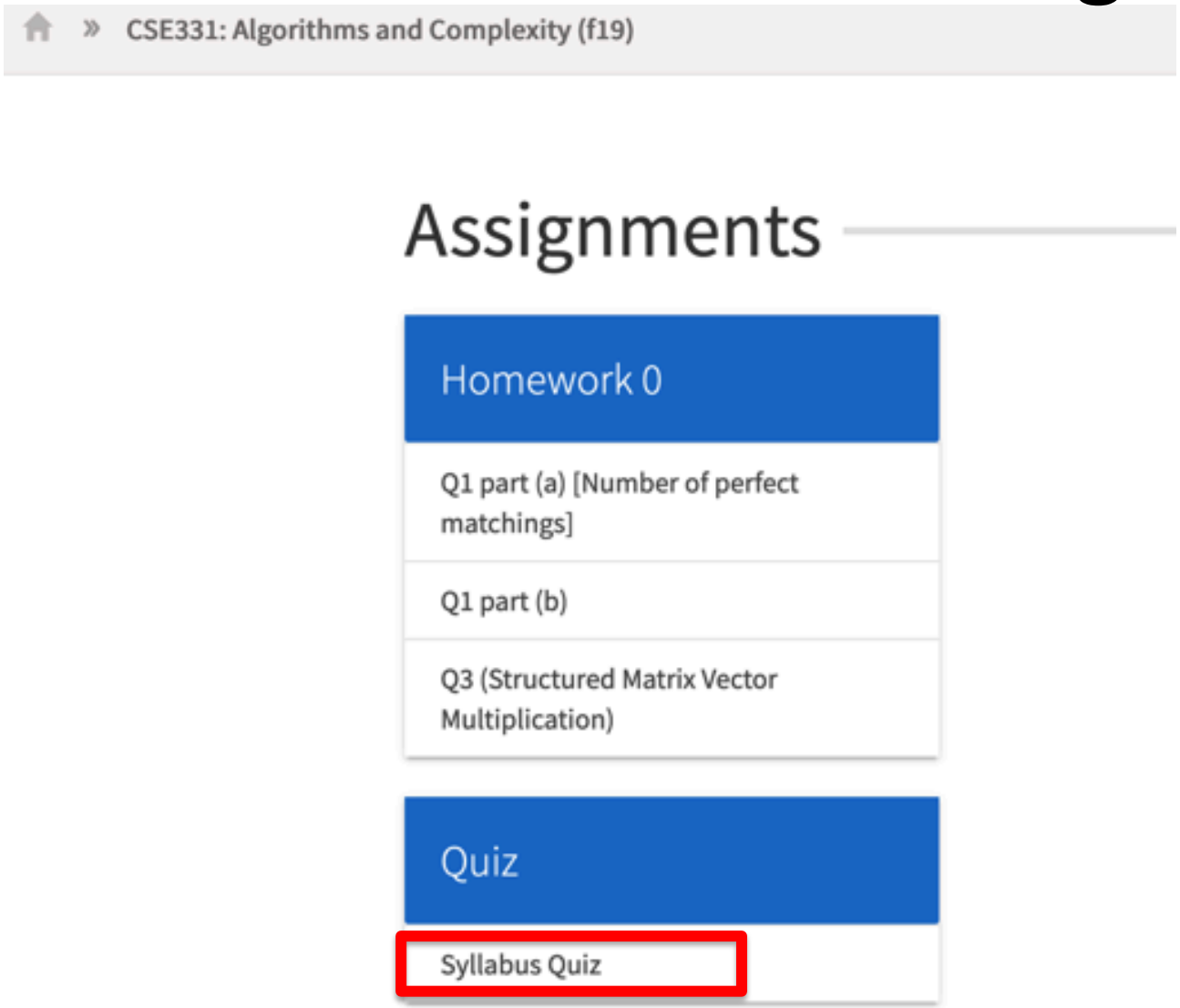
Once you have read the syllabus carefully, please fill in the Syllabus quiz on Autolab. As an incentive for you to fill in this form, **you will not receive any feedback on your assignments till you successfully answer AT LEAST 18 out of the 20 questions in the quiz**. (You can attempt the quiz as many times as you want.) Note that in addition to this syllabus, the quiz will also ask questions based on the homework policies.

Autolab



https://autograder.cse.buffalo.edu/

You can submit the following now



If you were registered by 9am on Monday, Aug 19 you should be on Autolab

Grading break-down

Grading Policy

Here is the split of grades:

Course Component	% of grade
Mini Project	10%
Homeworks	33%
Quizzes	3%
Exams	54%

Pre-requisites

Required (officially)

- CSE 250, [CSE 191 or MTH 311] and MTH 142
- At least a C-

Required (for practical purposes)

- Comfort with proofs
- Willingness to work hard!

Accessibility Resources

Information included in the syllabus

In short, let me know and consult with Accessibility Resources

Preferred Name

If you prefer using name diff from UB records

Let me know and we'll make a note of it.

Critical Campus Resources

Sexual Violence

UB is committed to providing a safe learning environment free of all forms of discrimination and sexual harassment, including sexual assault, domestic and dating violence and stalking. If you have experienced gender-based violence (intimate partner violence, attempted or completed sexual assault, harassment, coercion, stalking, etc.), UB has resources to help. This includes academic accommodations, health and counseling services, housing accommodations, helping with legal protective orders, and assistance with reporting the incident to police or other UB officials if you so choose. Please contact UB's Title IX Coordinator at 716-645-2266 for more information. For confidential assistance, you may also contact a Crisis Services Campus Advocate at 716-796-4399.

Mental Health

As a student you may experience a range of issues that can cause barriers to learning or reduce your ability to participate in daily activities. These might include strained relationships, anxiety, high levels of stress, alcohol/drug problems, feeling down, health concerns, or unwanted sexual experiences. Counseling, Health Services, and Health Promotion are here to help with these or other issues you may experience. You can learn more about these programs and services by contacting:

Counseling Services

- 120 Richmond Quad (North Campus), 716-645-2720
- 202 Michael Hall (South Campus), 716-829-5800

Health Services

Michael Hall (South Campus), 716-829-3316

Health Promotion

114 Student Union (North Campus), 716-645-2837

TA Office hours

YOU decide!

Recitations

Are on for this week!

Please stick to your recitation section

At least for the first month since all sections are full

Exams

Mid term (two parts)

Mon, **Oct 14** and Wed, **Oct 16**, 2019. Usual place and time.

Final exam

Fri, **Dec 13**, 2018. Norton 112, **12:00-2:30pm**

Things new to HWs in Fall 19

The dreaded Q3 is no more!

It survives as Q2 ☺

Q1 worth 50 points

The hard proof based Q2 and programming Q3 worth 25 points each

HWs due by 11:00am on Fridays

Other big change in Fall 18

Mini Project has two parts

As part of your mini-project you will consider societal implications of algorithms in two parts:

1. Make a `video` about ethical implications of an algorithm in real life; and
2. Do a `programming project` that involves making tradeoffs between various choices among which some have ethical dimensions.

See the tabs below for details on the two parts of the mini-project.

Acknowledgment

The development of this mini-project (especially the coding com
acknowledged.

Video Component | Coding Component

Coding Component

In the video component of the mini project y'all will study the ethical implications of an existing algorithm that has already been used in real life. In the coding component of the mini-project you will be developing and implementing algorithms in a scenario that is heavily influenced by a real life case study.

Your task

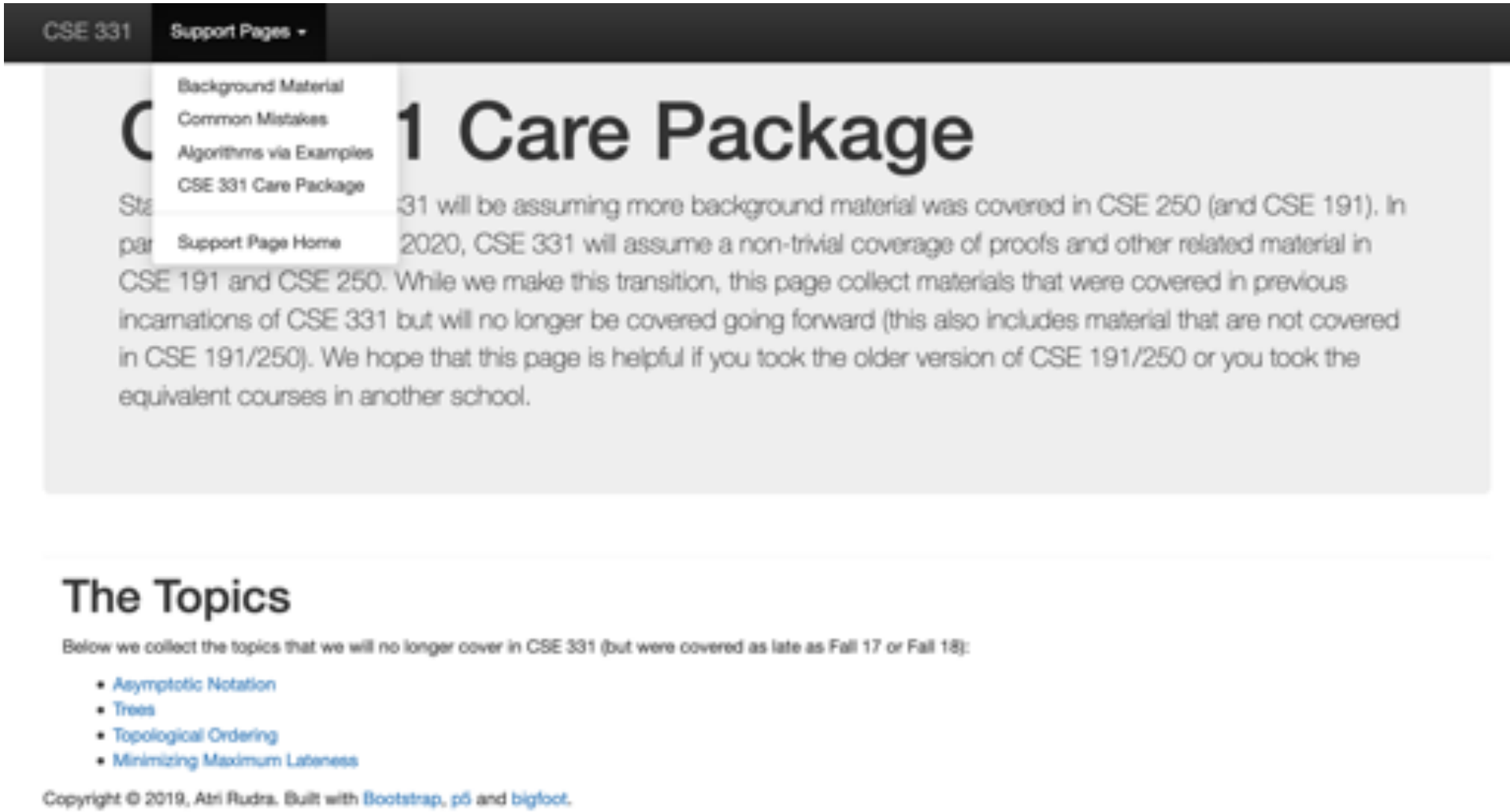
Your goal is to solve five programming problems and has to be done in **groups of size UP TO 3**. These will be setup in similar manner to programming question on the homeworks and will be due on Autolab.

How is this different from programming question on the homeworks?

Below we have listed some major ways in which this coding component will be different from programming questions on homeworks:

- First, unlike the HW programming questions, here you can collaborate with others. Also you will be allowed to refer to some external sources (more details forthcoming

Topic Coverage will change



C++ vs Java/Python

Use Java/Python if as you just as comfortable with as C++

Use a VM with g++ installed for Ubuntu

We recommend that you install a VM that runs `g++` on Ubuntu. In particular, we recommend that you use Jaric Zola's VM system that he created for his CSE 250 course.
If you have questions on Jaric's setup, please do **NOT** contact him: email `cse-331-staff@buffalo.edu` instead.

If you still prefer using your own system, we would still recommend that you test your code in the VM system above before submitting to Autolab.

poll ☆ 0 views

Help with installing VM for C++ (special office hours)

If you need some help installing the VM setup for submitting in C++:

http://www-student.cse.buffalo.edu/~atri/cse331/fall19/autolab.html

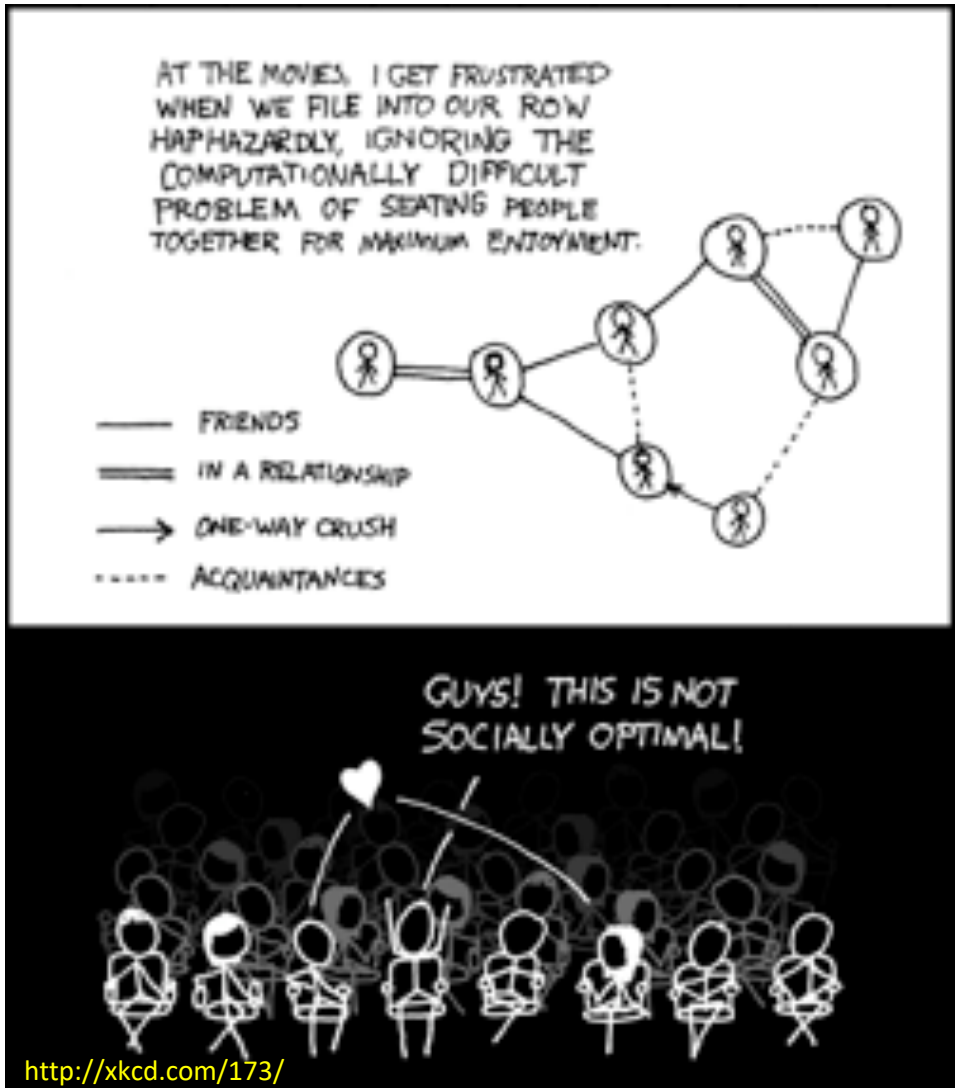
please select the times below that would work for you for some extra office hours. We will pick some of the most popular hour-long slots for Tuesday (August 27) and Wednesday (August 28). I would highly recommend that you try to install the vagrant system before you come to these office hours so that you can use the office hours to get help with trouble-shooting.

These slots are especially meant for transfer students who might not have had much practice working with linux/unix systems and have primarily use Windows as their main OS. Of course, these slots are open to anyone who needs help with this (even if you are not a transfer student!)

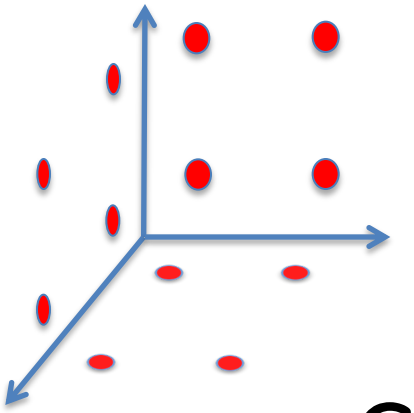
Please note that these office hours are just for help in installing the VM to run C++. The regular office hours will start from next week (do not forget to vote for those slots: @13).

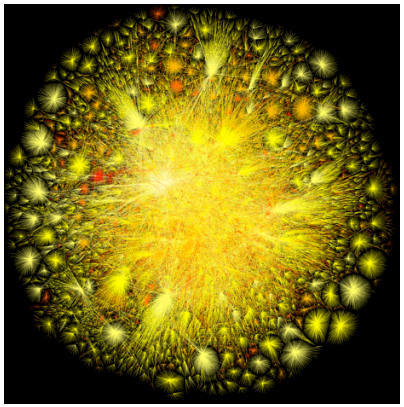
- Tue, 10:00-10:50am
- Tue, 1:00-1:50pm
- Tue, 2:00-2:50pm
- Tue, 3:00-3:50pm
- Tue, 4:00-4:50pm
- Tue, 5:00-5:50pm

This course: how to solve problems!

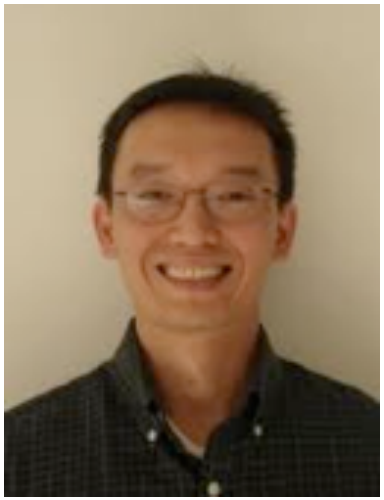


Why should I care ?





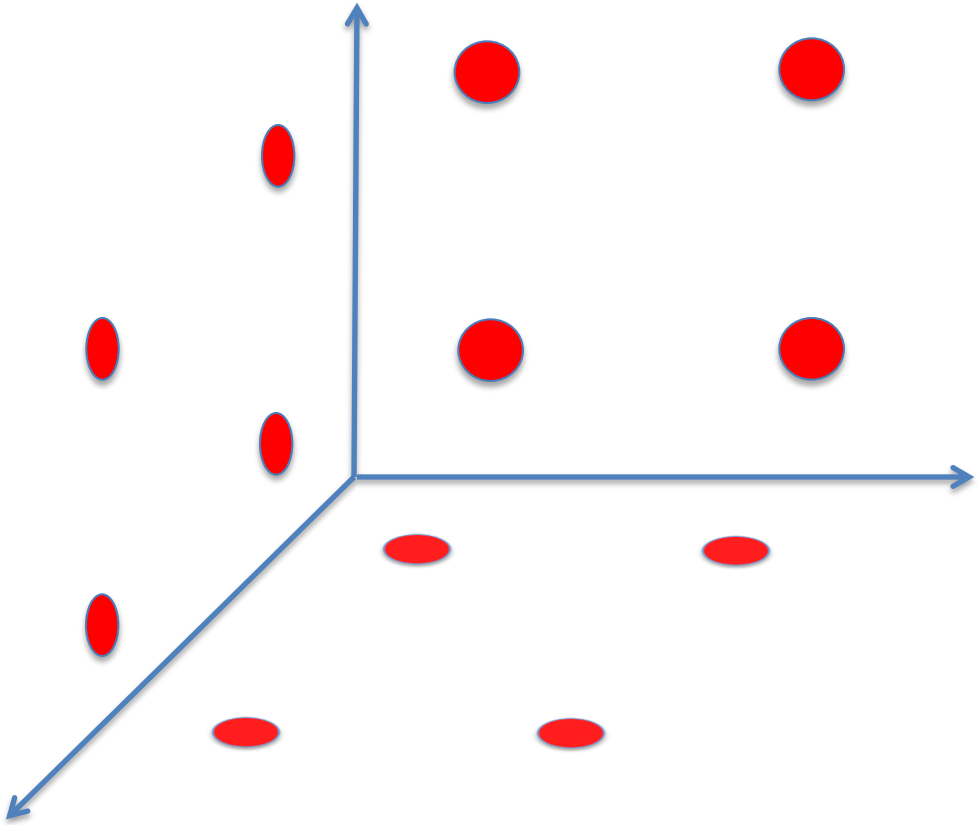
Combining Shadows to Understanding the network



LogicBlox

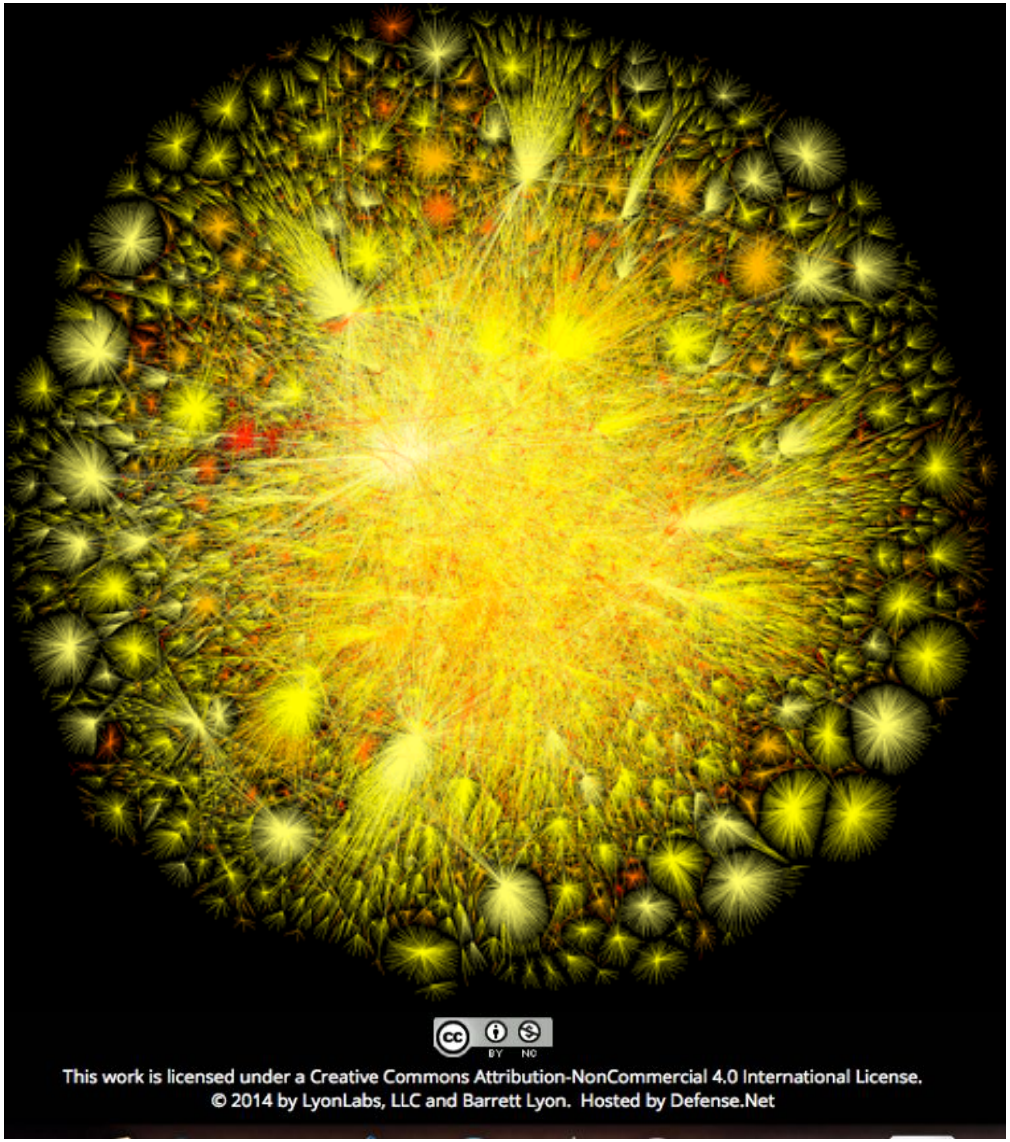
Stanford University

The key technical problem

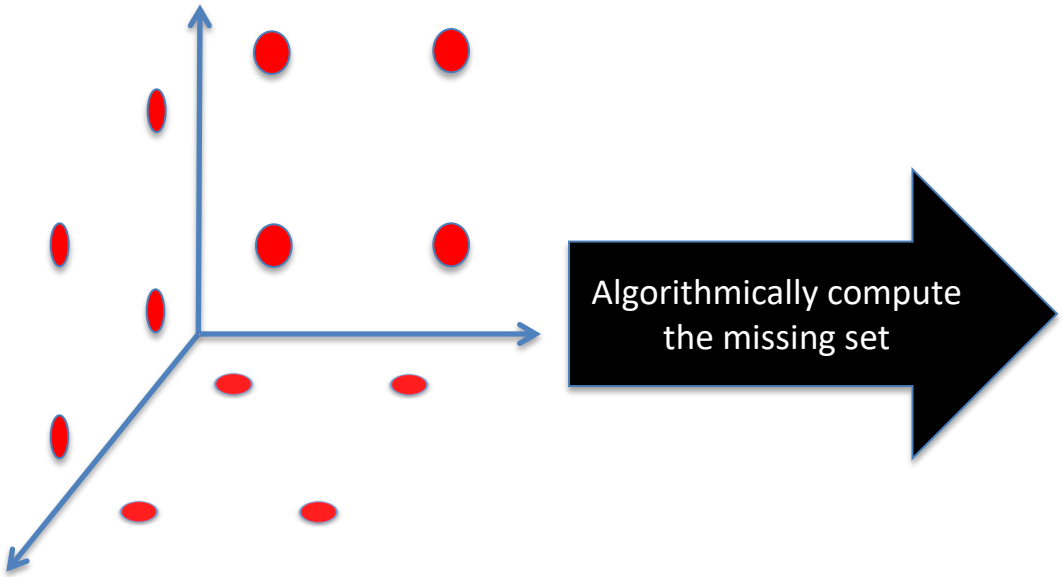


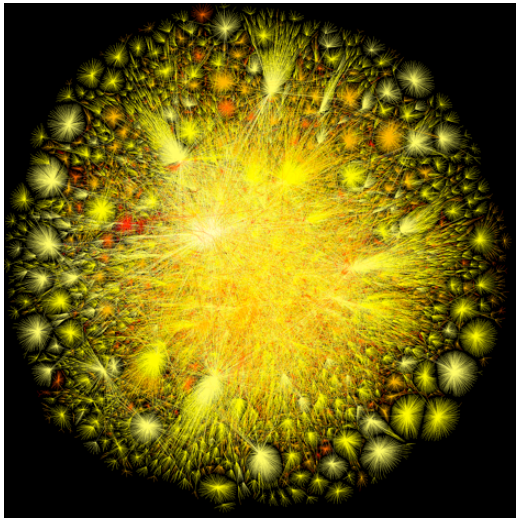
Given the three projections, what is the largest size of the original set of points?

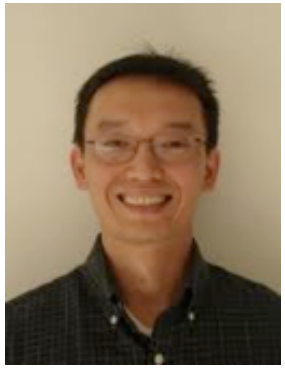
Detecting Communities



Conquering Shadows to Conquering the Internet







The proof is in the performance

EMPTYHEADED




10x faster

ORACLE

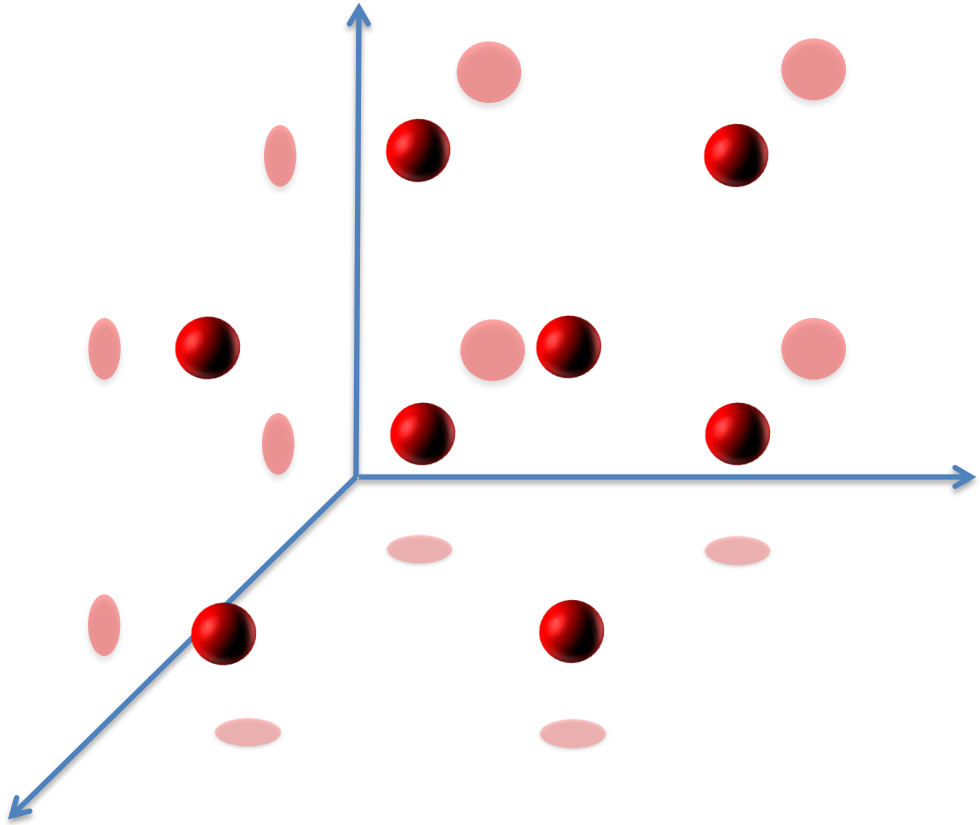
Dato GraphLab

LogicBlox

SNAP

Better algorithm with little hacking will beat a worse algorithm with tons of hacking

The key technical problem



Highly trivial: $4^3 = 64$

Still trivial: $4^2 = 16$

Correct answer: $4^{1.5} = 8$

If detecting communities is not for you

From someone who got a Google job

“You can let your algorithms class know that the phone interviews are essentially like **a difficult algorithms test**.

Lots of data structures, specifying the algorithm, analyzing the run time and space requirements... And all on the phone and **you're supposed to talk through your thought process**.”

Coding jobs will be done by AI

stacksort

In a recent xkcd's alt text, Randall Munroe suggested **stacksort**, a sort that searches StackOverflow for sorting functions and runs them until it returns the correct answer. So, I made it. If you like running arbitrary code in your browser, try it out.

Like (or hate) it? Comment on HackerNews

stackoverflow_sort(
Try a list of numbers, a string, a list of words or json.

[8,6,7,5,3,0,9]

);

Sort

var output =
Output from the function.

;

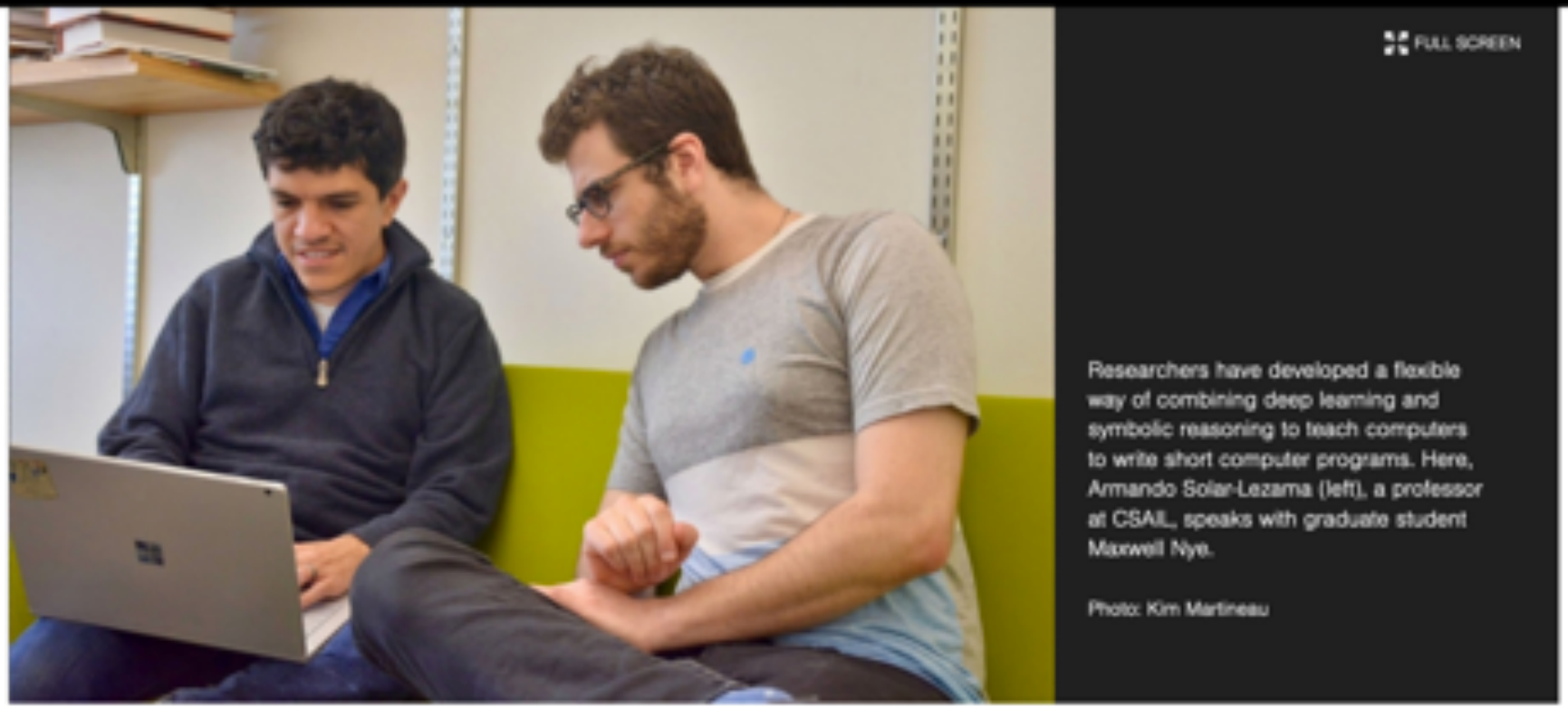
output console

Coding jobs will be done by AI

MIT News

ON CAMPUS AND AROUND THE WORLD

Browse or Search



FULL SCREEN

Researchers have developed a flexible way of combining deep learning and symbolic reasoning to teach computers to write short computer programs. Here, Armando Solar-Lezama (left), a professor at CSAIL, speaks with graduate student Maxwell Nye.

Photo: Kim Martineau

Toward artificial intelligence that learns to write code

Researchers combine deep learning and symbolic reasoning for a more flexible way

So am I doomed?

There will still be room for high level *algorithmic* thinking!

European Association for Theoretical Computer Science

HOME ABOUT SEARCH CURRENT ARCHIVES

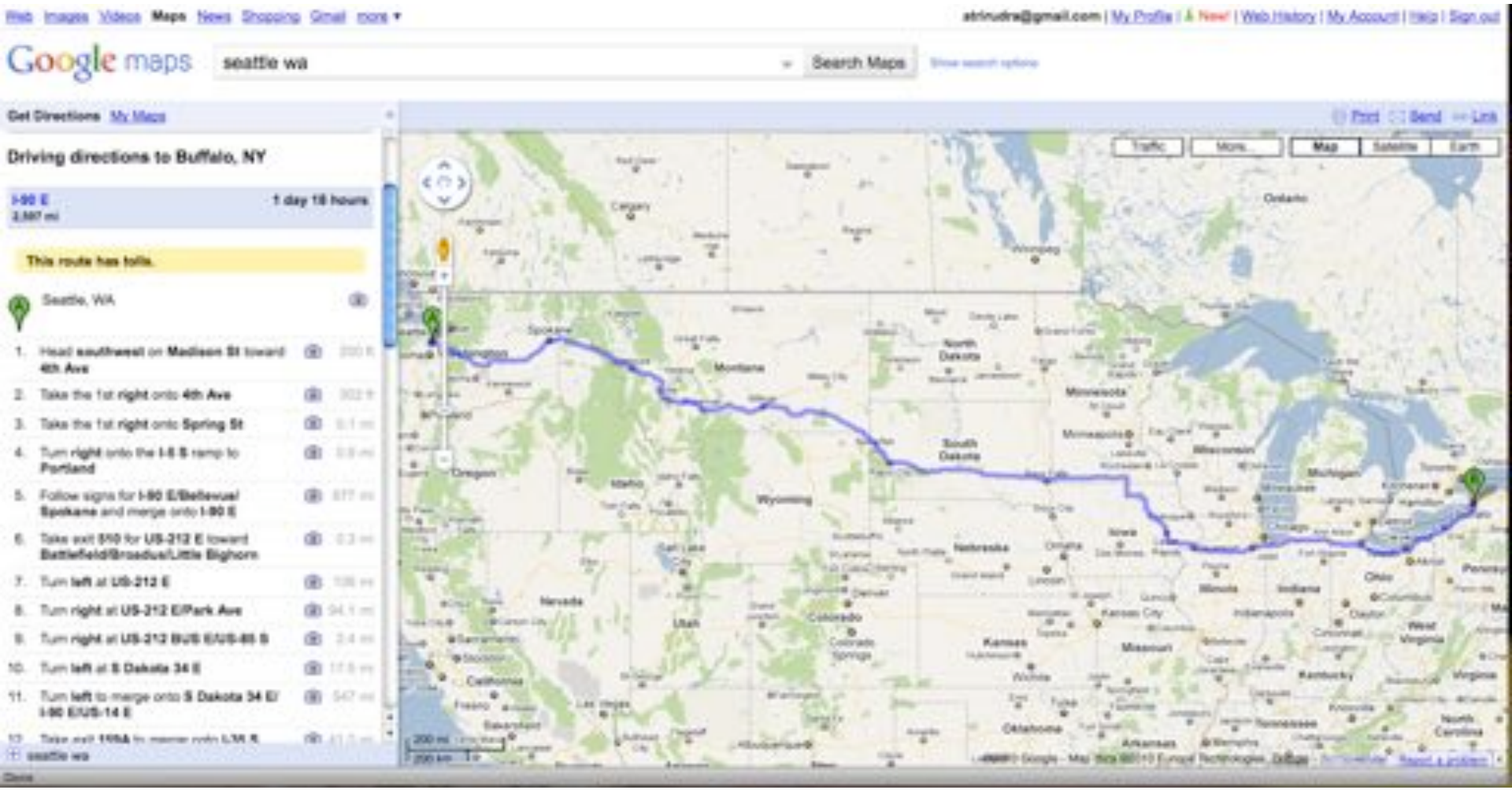
Home > No 125: June 2018 > **Lamport**

If You're Not Writing a Program, Don't Use a Programming Language

Leslie Lamport, Distributed Computing & Education Column by Juraj Hromkovic, Stefan Schmid

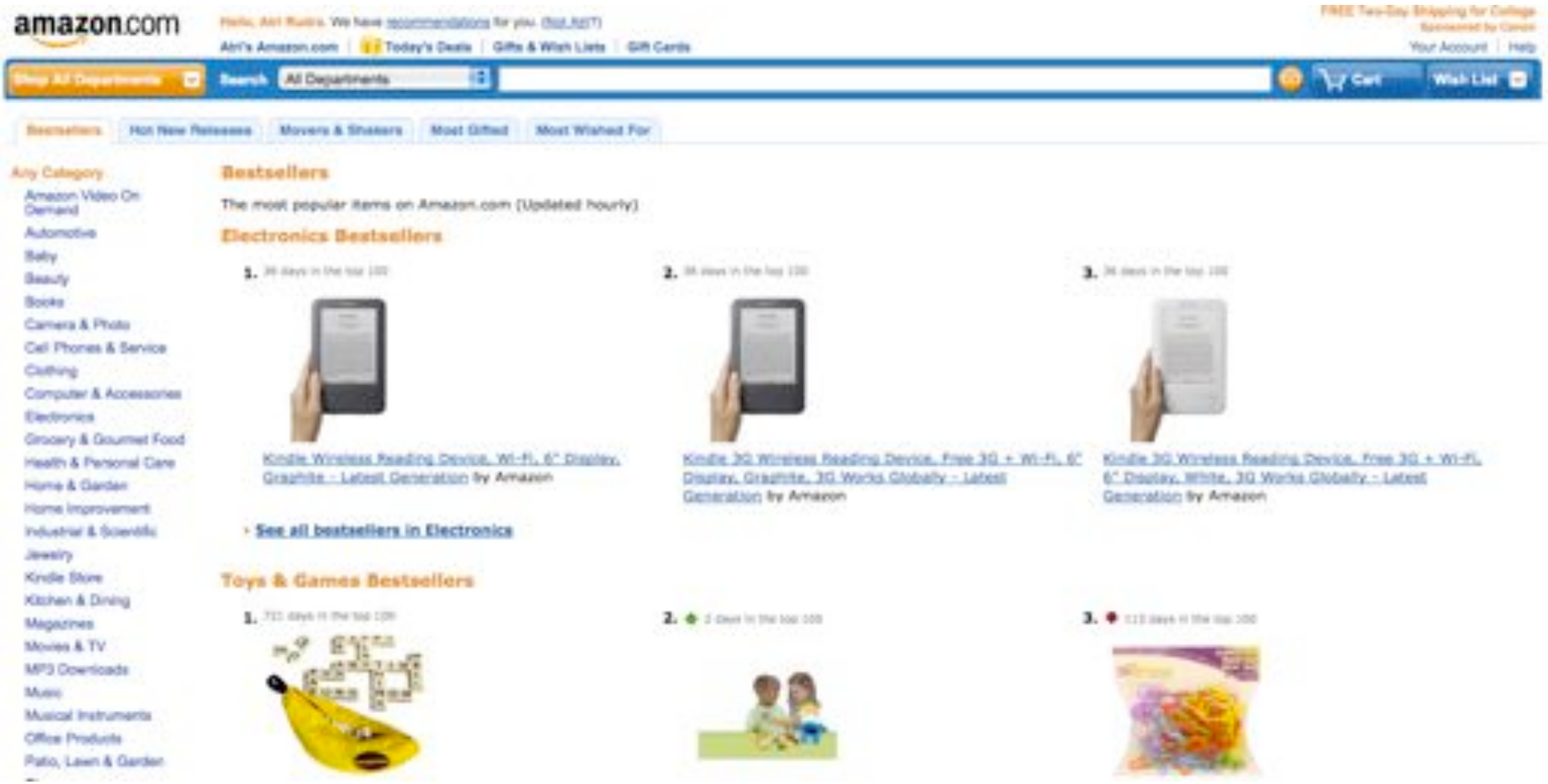
Today, programming is generally equated with coding. It's hard to convince students who want to write code that they should learn to think mathematically, above the code level, about what they're doing. Perhaps the following observation will give them pause. ***It's quite likely that during their lifetime, machine learning will completely change the nature of programming. The programming languages they are now using will seem as quaint as Cobol, and the coding skills they are learning will be of little use. But mathematics will remain the queen of science, and the ability to think mathematically will always be useful.***

Why care about algorithms?



Driving directions

Why care about algorithms?

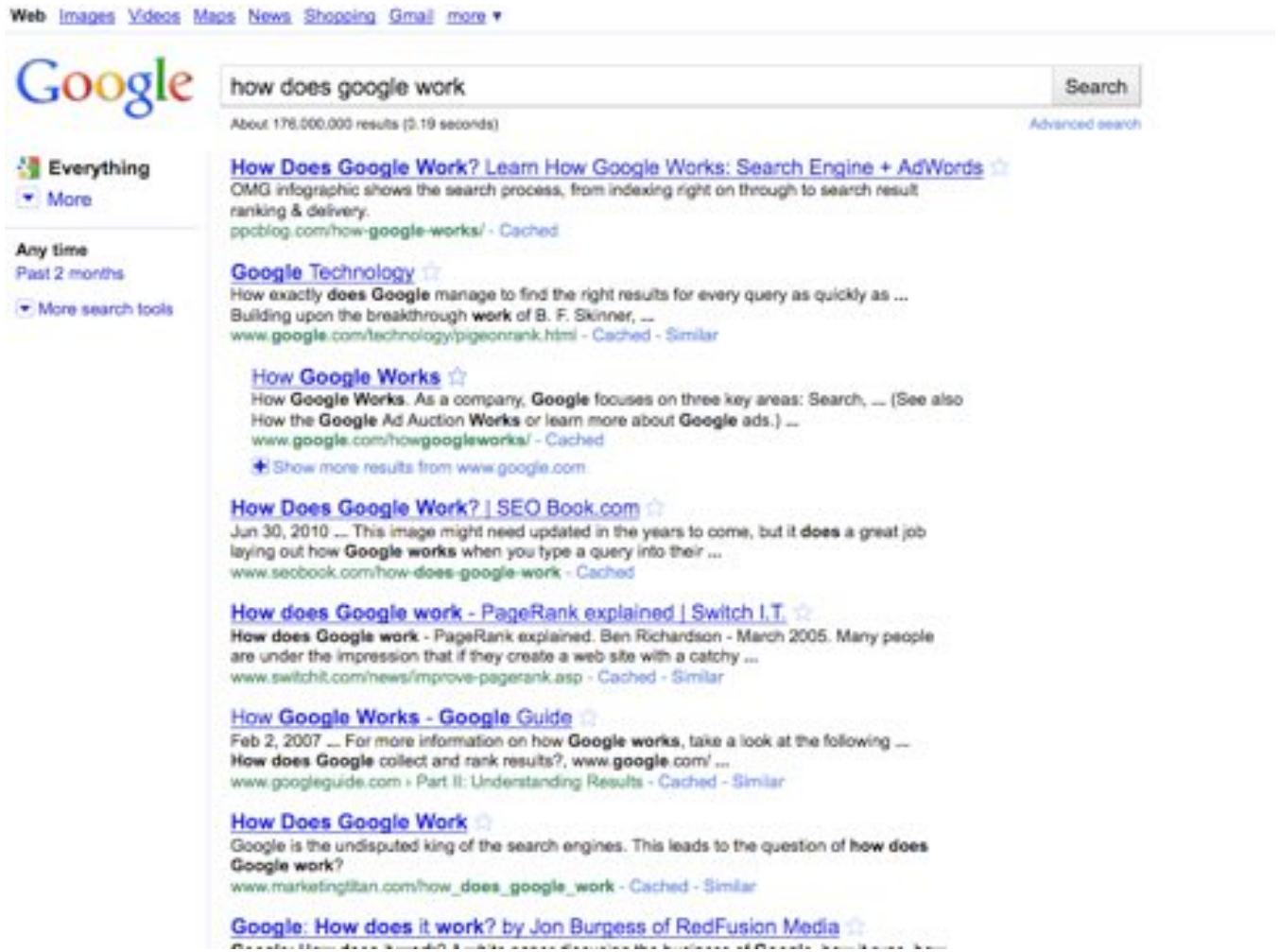


Computing Bestsellers on the fly

Why care about algorithms?

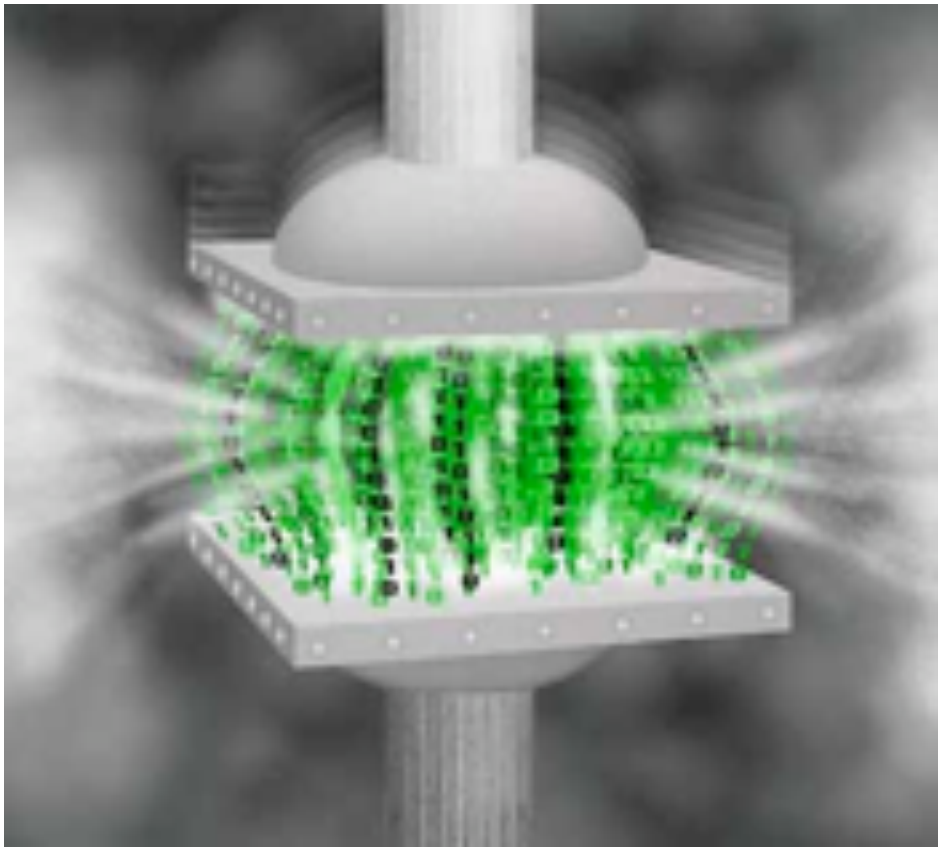
Booking cheapest air tickets

Why care about algorithms?



Google searches

Why care about algorithms?



http://www.di.ens.fr/~cherniav/teaching.html

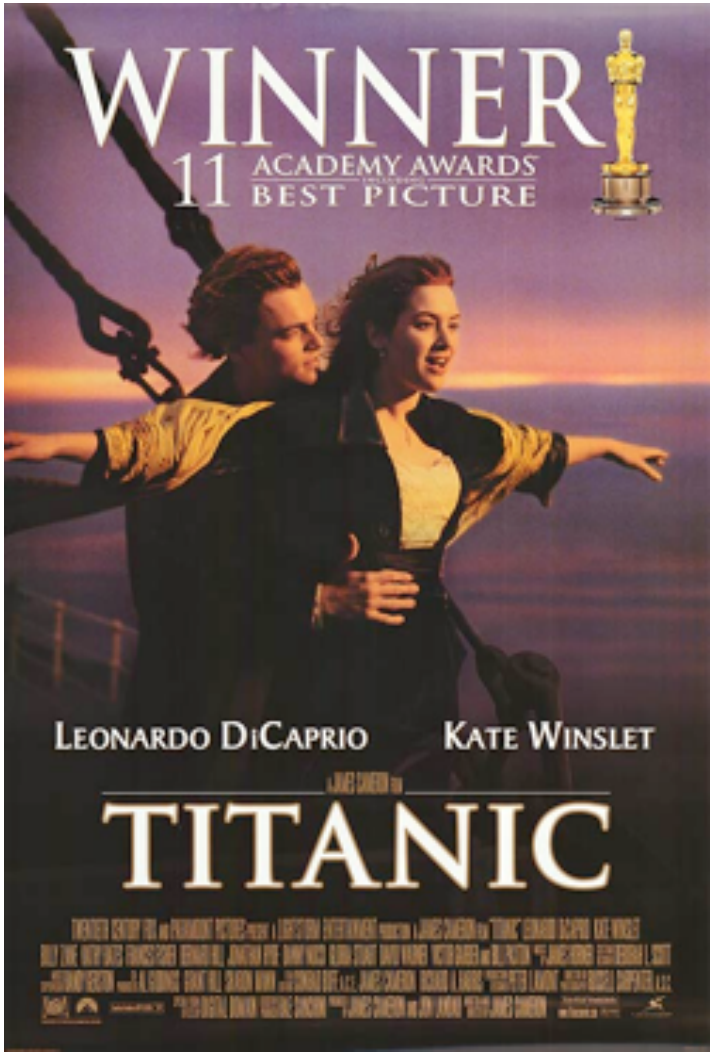
Data compression

Why care about algorithms?

http://www.switched.com/2010/02/11/fix-dvd-scratches-using-a-banana-and-toothpaste/ courtesy: Unplgd

Error correction

(And I could) go on…



http://www.movieposter.com/poster/MPW-33672/Titanic.html

Find out for yourself

Mini project: Video on ethical impacts of algorithm. Groups of size = 3

CSE 331 Mini Project

Fall 2019

Details and motivations for the mini project.

Under Construction

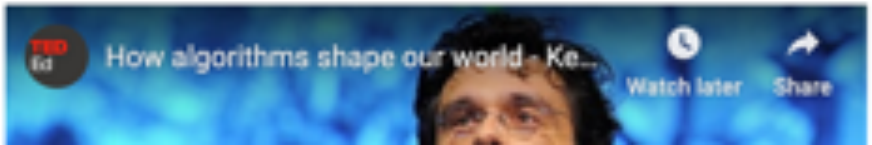
This page is still under construction. In particular, nothing here is final while this sign still remains here.

Motivation

CSE 331 is primarily concerned with the technical aspects of algorithms: how to design them and then how to analyze their correctness and runtime. However, algorithms are pervasive in our world and is common place in many aspects of society. The main aim of the mini-project is to have you explore in some depth social implications of algorithms.

Just to give two examples for such implications:

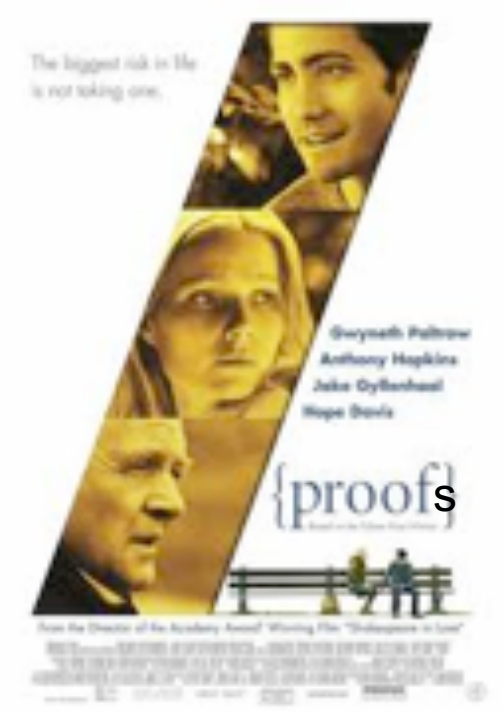
- Algorithms are pervasive in financial transactions and these algorithms have consequences beyond just trading:



Questions/Comments?

Now about the course

We'll do loads of



http://www.impawards.com/2005/proof.html

Writing down your thought process formally and precisely!

An incorrect “proof”

A more subtle incorrect “proof”

Brad Pitt had a beard

waleg.com

Every goat has a beard

animaldiversity.org

Hence, Brad Pitt is a goat.

Why do proofs?

Makes you think logically about problems and solutions

From a friend who works on Google Maps:

> Proving that the algorithm I am implementing is correct helps me identify corner cases

CSE 331 Support Pages

Why should we do proofs?

We will focus a lot on proofs in CSE 331. In this document I will motivate why doing proofs is good even though you might not do proofs for a living. While doing this, we will also go through examples of how to write algorithm ideas and details as well as proof ideas and details (which you will need to write in your homework solutions).

Some reasons to do proofs

In this section, I will lay out some reasons why I think it is beneficial for you guys to do proofs. The first two are probably more along the lines of "if you do proofs for a living" situation. The rest of the reasons should be valid for all of you. I will try and make the reasons as concrete as possible: in the next section, we will consider algorithms for the specific problem of generating all permutations (recall that we previously had punted on designing an algorithm for this problem).

Sometimes you might not have a choice

One of the easiest way to verify an algorithm idea you have is to code up the algorithm and then test it on some (say random) inputs. However, sometimes this might not be a choice. E.g. if you work on Quantum Computing, then you do not have a quantum computer to run your quantum code on! So currently pretty much the only choice you have is to *prove* that your algorithm is indeed correct. For example, one of the crowning achievements of quantum computing is Shor's algorithm to computes the factors of large numbers efficiently on a quantum computer (that recall does not exist yet!). (You might also want to read Scott Aaronson's high level description of Shor's algorithm.) The reason why factoring large numbers is important is that if one can solve this problem efficiently then one can break the RSA cryptosystem. RSA is used everywhere (e.g. when you use your credit card online, RSA is used to make the transaction secure), so this is a big deal.

http://www-student.cse.buffalo.edu/~atri/cse331/support/proofs/index.html

A common complaint

Your examples in class look nothing like HW questions.

True because….

zazzle.com

False because…

HWs and exams will test your **understanding** of the material

To get an A in the class

Have to get at least 90.000000000000000000000%

Rest graded on the curve

A cautionary tale…

When I was an undergrad

Took algorithms as a sophomore

Understood all the lectures

Did not study outside of lectures

(We had no homeworks)

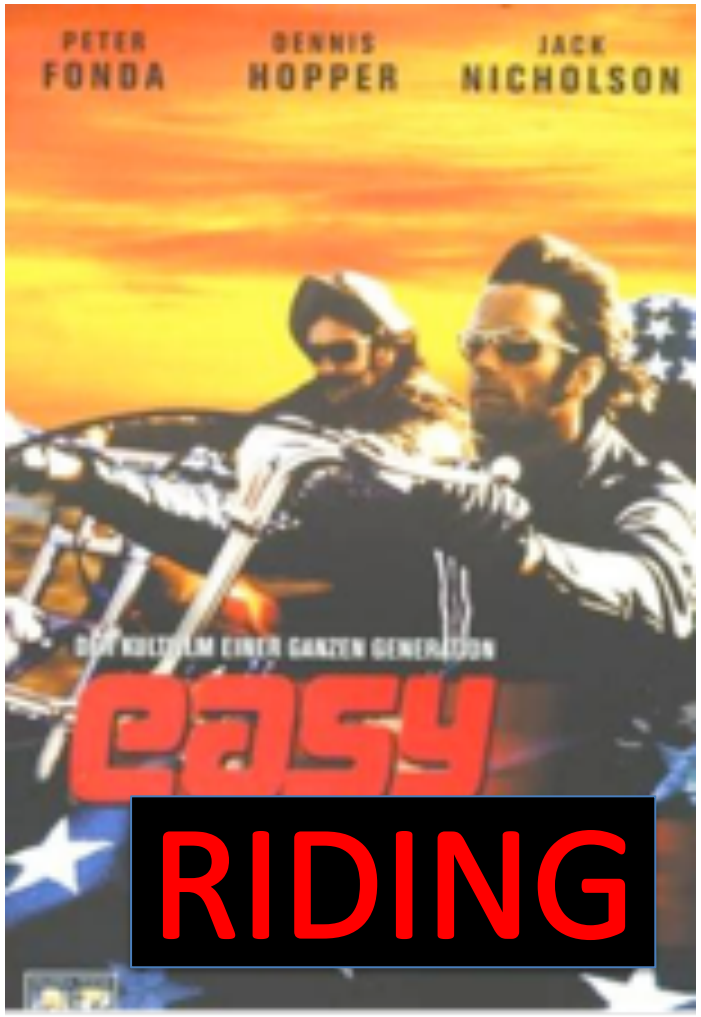
Did decent on the mid-term

Nearly flunked the finals

Got a **C**

Questions/Comments?

How we will make 331



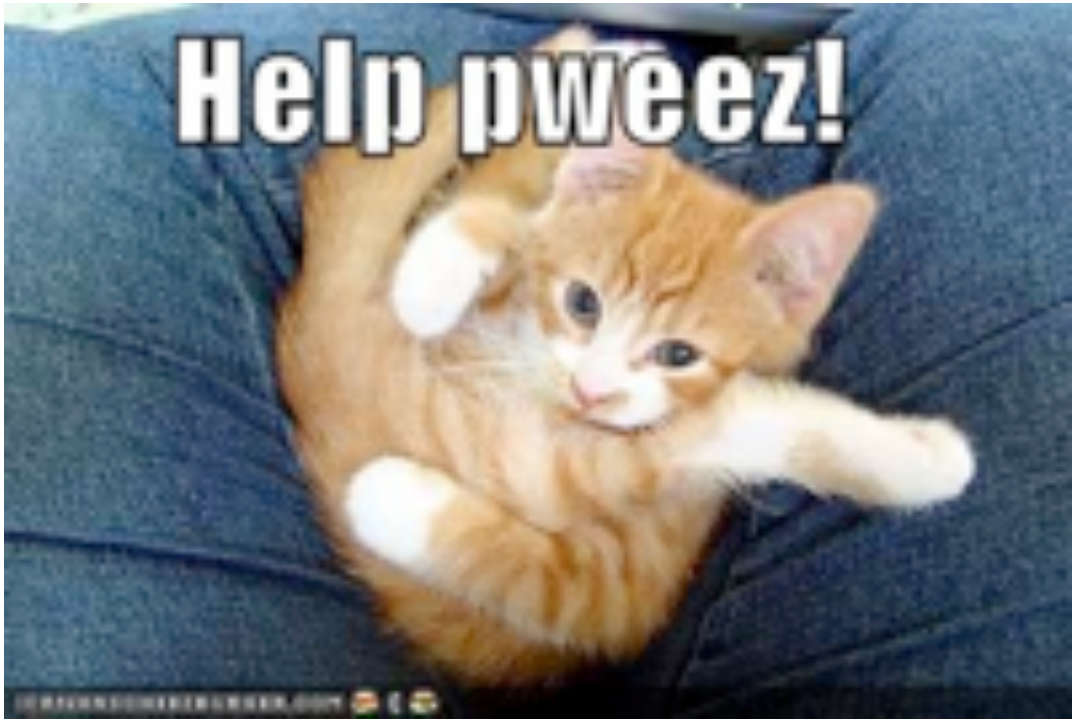
What we'll strive to do

Help you with your questions and/or doubts

If need be, email us for time outside of regular office hours

We’re not mind readers

If you need it, ask for help



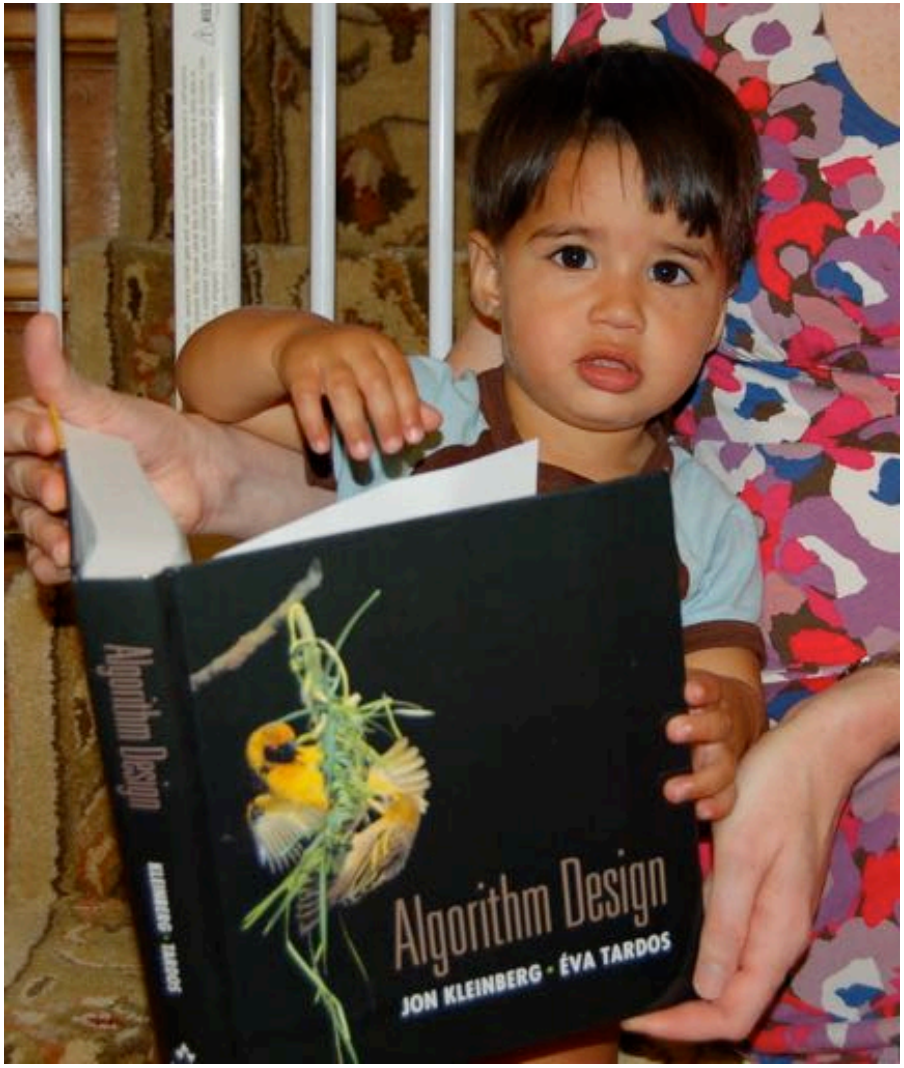
More chances to recover

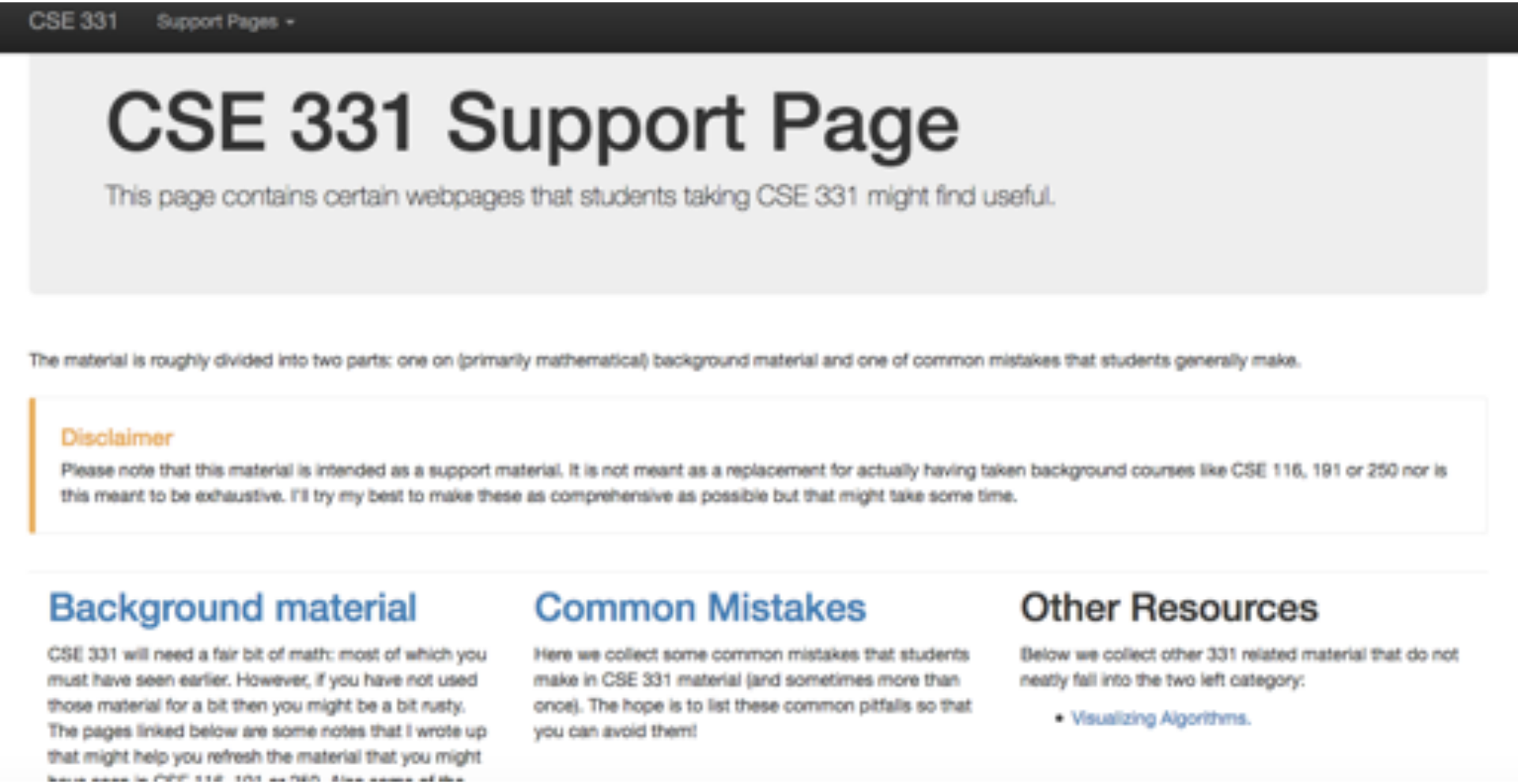
Lowest three HW scores will be dropped

If you do better on the final exam than the mid-term exam

then only final exam score will count

Follow the Textbook





http://www-student.cse.buffalo.edu/~atri/cse331/support/index.html

The cautionary tale has asilver lining…

C in undergrad algorithms

Ph.D. in algorithms/complexity

The only way to do well is to work hard