Lecture 40

CSE 331

Dec 11, 2023

Final exam post

note @505 stop following 38 views Actions

Final exam post

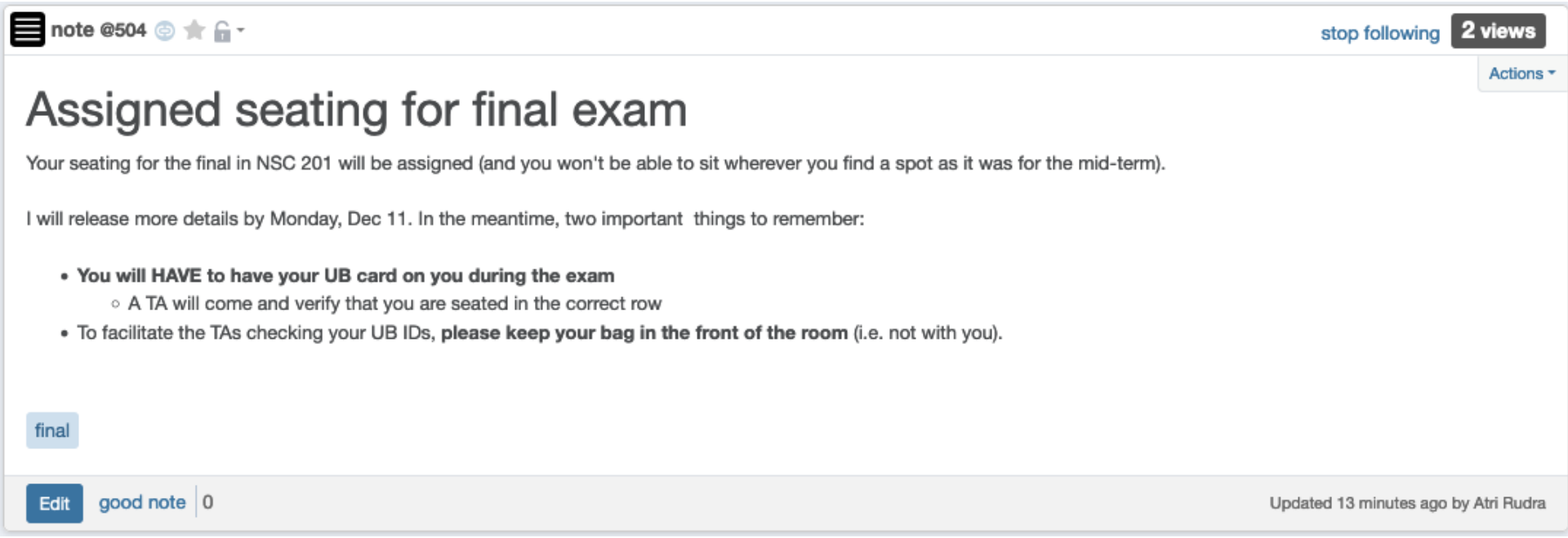
I'll start off with some generic comments:

- The final exam will be based on all the material we will see in class up to NP-completeness of k-colorability (we'll finish that stuff by either Friday, Dec 8 or Monday, Dec 11).
  - In case you want a head-start we will cover Sections 8.1-8.4 and Section 8.7 in the textbook. For the rest the schedule page details what sections of the book we have already covered.
  - I know this does not give a huge lead time into the final exam but unfortunately since we are running one lecture behind previous years means less lead time than in previous years.
- Exam will be from **12:00pm to 2:30m** on Wednesday, **Dec 13** in class (**NSC 201**). Note that the exam will be for 2.5 hours and *not 3 hours* as it says on HUB.
  - ~~I will post the makeup final exam time (due to exam conflicts-- see @447) by Monday, Nov 20.~~ If you had emailed me about a conflict with the final exam time (see @447), I have emailed you back the details on the timing of the makeup final exam.
- **DO NOT FORGET TO BRING YOUR UB CARD TO THE EXAM** (@504)

Next are comments related to **preparing for the finals**:

1. Take a look at the sample final (@503) and spend some quality time solving it. Unlike the homeworks, it might be better to try to do this on your own. Unlike the sample mid-term, this one is an actual 331 final exam so in addition to the format, you can also gauge how hard the final exam is going to be (your final exam will be the same ballpark). However as with the sample mid-term, you make deductions about the coverage of topics at your own peril (but see points below). Once you have spent time on it on your own, take a look at the sample final solutions (@503).
2. The actual final will have the same format as the sample final: The first question will be T/F, 2nd will be T/F with justification, the rest of the three will be longer questions and will ask you to design algorithms (parts of them might be just *analyzing* an algorithm.)
3. For the T/F questions (i.e. the first two questions), anything that was covered in class or recitations or piazza is fair game. If you want to refresh your memory on what was covered, take a look at the schedule page. If you want quick summaries of (almost all) the lectures, review the lecture notes or slides or videos.
4. To get more practice for the T/F questions, review all the T/F polls on piazza (@60)
5. For the remaining 3 questions, one will be on greedy algorithms, one will be on divide and conquer algorithms and one will be on dynamic programming. However, note that Chapter 2 and 3 in the book are basic stuff and almost any question in the final could fall under the purview of those two chapters. There will be **at least** one T/F and one T/F with justification Q for the NP-complete material so y'all should definitely focus on those as well but I will not ask any "proof based" Qs on that material.
6. In previous finals, like your mid-terms, there have been questions that are either straight lifts from homeworks or are closely related and this trend will continue in the actual exam (though to a lesser extend then the mid-term). This means that you should review your homeworks (all of them) before the exam. Also make sure to review the support pages and recitation notes.
7. If you are short on time and you are prioritizing the topics to study, keep points 5 and 6 above in mind.
8. Sections in the book that were not covered at all in the class but were handed out as reading assignments or recitation notes: I can also ask any direct questions from them. In addition, it might be useful to read them to get a better feel for the material. In any case once you have read the material covered in class a couple of times, it might do your brain some good to read some different material.
9. You can bring in two 8.5"X11" review sheets (you can use both sides on both). Use this judiciously: they can be a very useful tool to note down some weird things you have a hard time remembering and/or noting down specific

Bring UB card to final exam!



Details on seating assignment by tomorrow 12pm

Project Survey Out!

note @561 stop following 0 views Actions

Project survey now open!

As a reminder that in addition to P4+5 coding and reflection problems, y'all all need to fill in a survey.

The survey was originally supposed to go out Friday at noon but I decided to release it earlier just in case if there are any issues, there is enough time to fix.

The instructions are at the bottom of the survey page. The only place where I can potentially see issues happening is if I uploaded incorrect group information (unlikely but possible). If so, please let me know ASAP.

Note: If a group member resigned the course they would still show up and this is not a bug. Make sure you give them all 0s so that you get credit for working in a smaller group.

Also note that you will rate yourself AND two other group members.

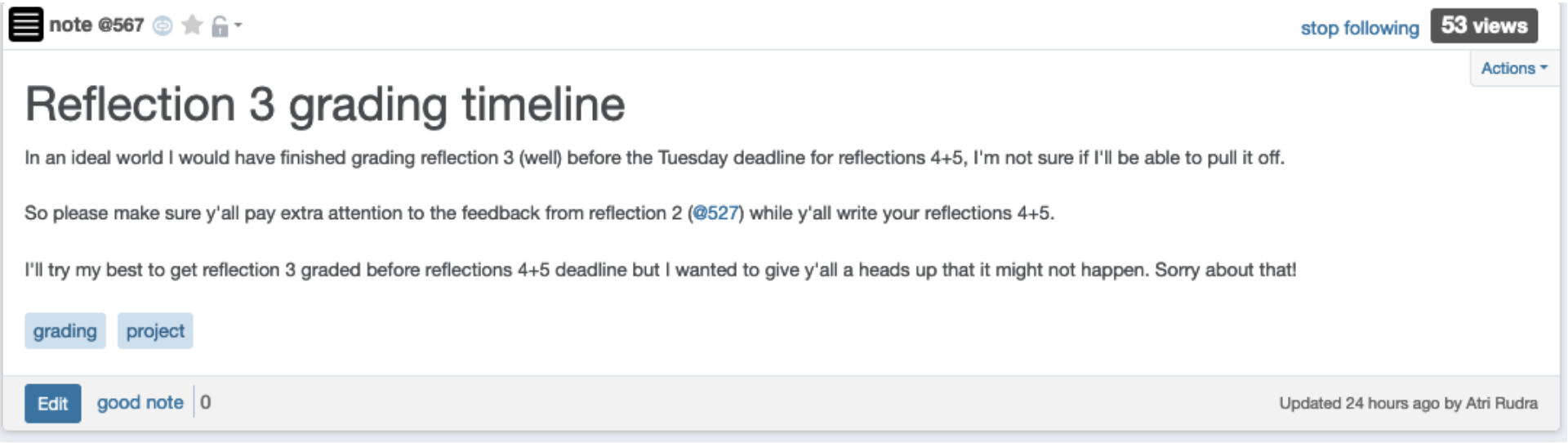
I do not control the actual submission site so sooner I get bug reports the better! In particular, *if I get a bug report after Friday 5pm, I cannot guarantee any fixes before the deadline*. Note that I do **not** expect there to be bugs but some changes were made recently to the website and I'm just being cautious here!

So please check out the system at your earliest convenience and if you spot any issues, please report back in the comment section below-- thanks!

project

Edit good note | 0 Updated 43 seconds ago by Atri Rudra

Timeline on reflection 3 grading



Extended Trevor OH tomorrow

note @571

71 views

Actions

Additional office hours for the final

Hey everyone,

In an attempt to alleviate some of the stress from finals, I am going to hold some extended office hours specifically for the final on **Tuesday, December 12th**. This will be a time where I'll do my best to answer any lingering questions that you may have or give explanations of content from the course that you may have not fully understood. Details below:

- **Where:** Conference room in Capen 212 (Take the elevator next to 1 Capen to the 2nd floor, turn right. I'll have the door open and hopefully a sign.)
- **When:** Tuesday, December 12th starting at 10am until whenever I feel like it (usually a few hours).

Hope to see many of you there!

- Trevor

final office_hours

~ An instructor (Mokshita Gupta) endorsed this note ~

Edit good note 3

Updated 2 days ago by Trevor Schneggenburger

Now relax…

Halting Problem

Input: A program P and input I

Output: Yes if P terminates on I
No otherwise

Theorem: There is no algo A' for Halting problem (as long as A' terminates)

Let us assume a `magic_box` for A' exists!

```
def magic_box ( P, I):
# This is a magic box so there is no real code here!
''' Return True if P halts on I and False otherwise'''
```

Assume that

1. `magic_box` terminates on all inputs
2. `magic_box` ALWAYS correctly decides if P halts on I or not

A new function contradiction

```
def contradiction ( P ): # This function takes a program as
an input

#Run magic_box on (P,P)
if magic_box (P,P): # Use an UTM to make this call
   while True:
      pass # Do nothing

return # Just terminate if magic_box(P,P) returns False
```

Since we assumed `magic_box` exists `contradiction` is well defined!

A function call

```
contradiction (contradiction) # Use an UTM to make this call
```

Since we assumed `contradiction` is well defined, the above is a legit function call!

```
contradiction
(contradiction)
```

What are outcomes of the function call `contradiction (contradiction)`?

It terminates

It does not terminate

Case 1:

contradiction (contradiction)
terminates

```
def contradiction ( P ): # This function takes a program as an i

#Run magic_box on (P,P)
if magic_box (P,P): # Use an UTM to make this call
    while True:
        pass # Do nothing

return # Just terminate if magic_box(P,P) re
```

This does NOT terminate!!

What should this call return?

```
contradiction (contradiction ):

if magic_box (contradiction, contrac
    while True:
        pass # Do nothing

return
```

We get into an infinite loop here!

Case 2:

contradiction (contradiction) does not terminate

```
def contradiction ( P ): # This function takes a program as an i

#Run magic_box on (P,P)
if magic_box (P,P): # Use an UTM to make this call
    while True:
        pass # Do nothing

return # Just terminate if magic_box(P,P)
```

```
contradiction (contradiction ):

if magic_box (contradiction, contrad
    while True:
        pass # Do nothin

return
```

This DOES terminate!!

What should this call return?

We return here!

Let's recap: Argue `magic_box` doesn't exist

1. Assume `magic_box` exists
2. Defined a function `contradiction` that uses `magic`
3. Looked at the ONLY two possibilities

Contradiction!!

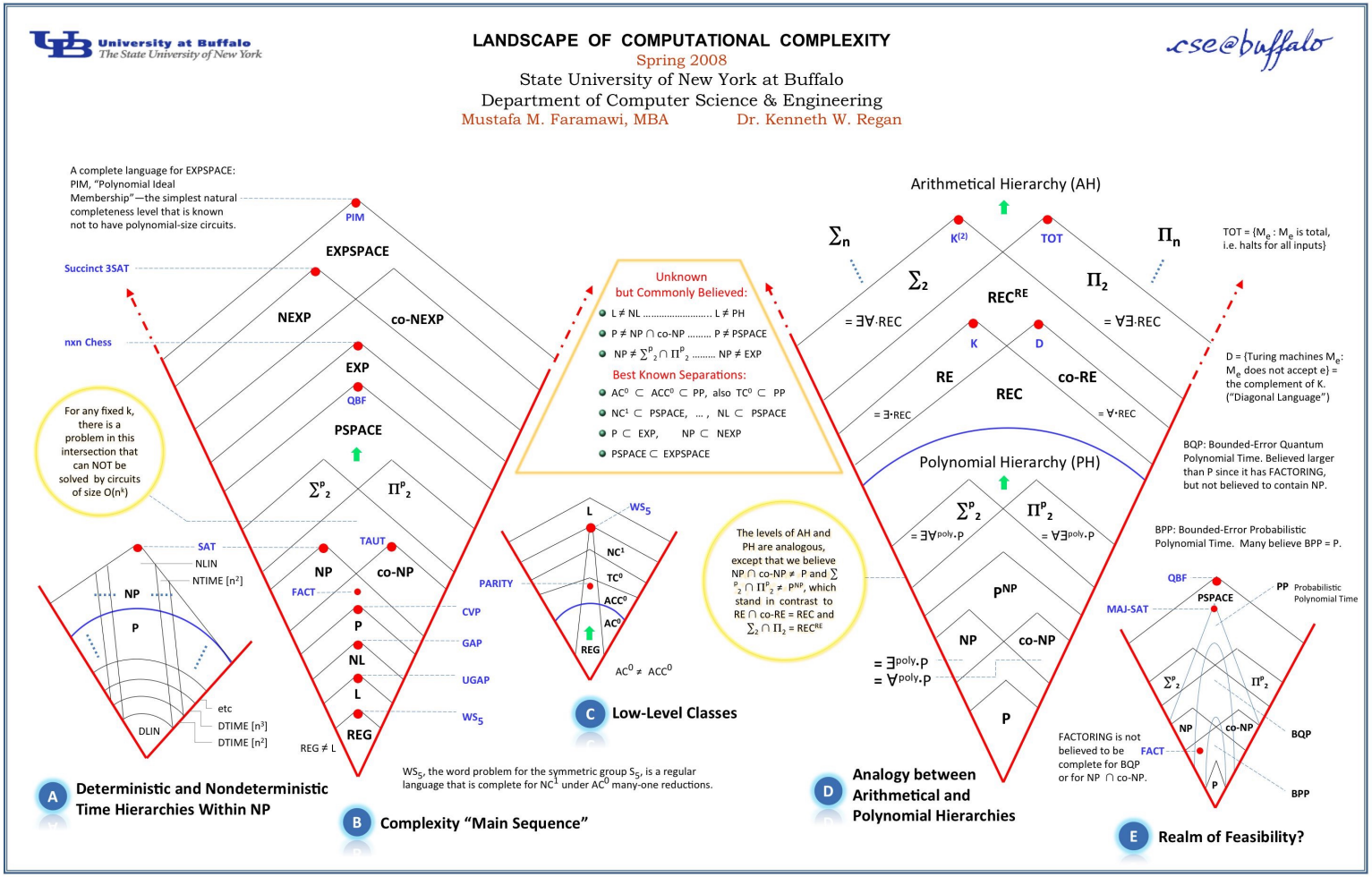
3.1. `contradiction(contradiction)` terminates
→ `contradiction(contradiction)` does NOT terminate

3.2. `contradiction(contradiction)` does NOT terminate
→ `contradiction(contradiction)` terminates

Theorem: There is no algo A' for Halting problem (as long as A' terminates)

Questions?

Anything > NP and < undecidability?



Randomized algorithms

What is different?

Algorithms can toss coins and make decisions

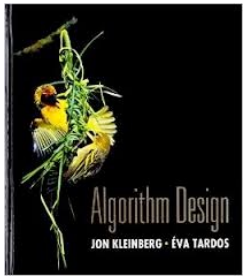
http://calculator.mathcaptain.com/coin-toss-probability-calculator.html

A Representative Problem

Hashing

Further Reading

Chapter 13 of the textbook



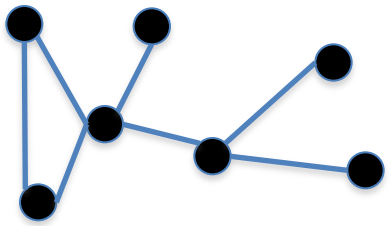
CSE 432

Approximation algorithms

What is different?

Algorithms can output a solution that is say 50% as good as the optimal

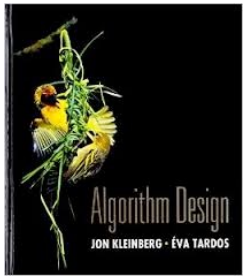
A Representative Problem



Vertex Cover

Further Reading

Chapter 12 of the textbook



Online algorithms

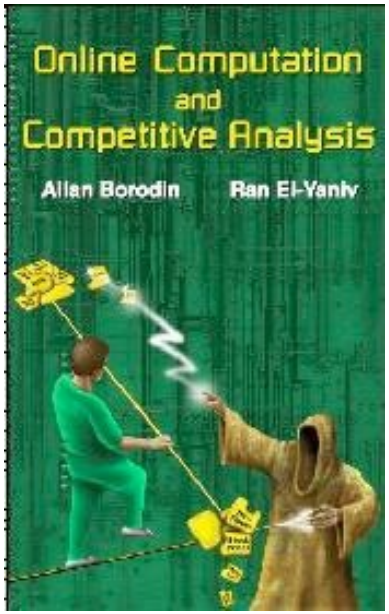
What is different?

Algorithms have to make decisions before they see all the input

A Representative Problem

Secretary Problem

Further Reading



Data streaming algorithms

https://www.flickr.com/photos/midom/2134991985/

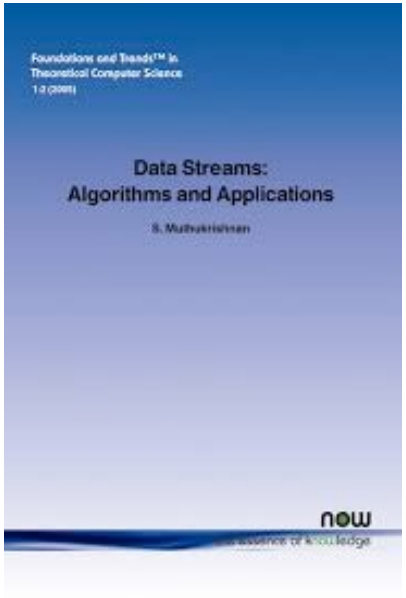
What is different?

One pass on the input with severely limited memory

A Representative Problem

Compute the top-10 source IP addresses

Further Reading



Distributed algorithms

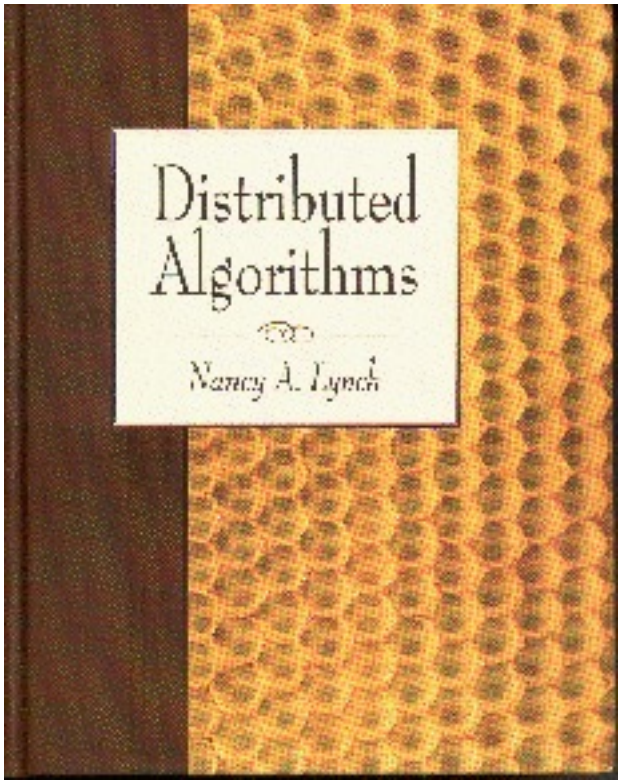
What is different?

Input is distributed over a network

A Representative Problem

Consensus

Further Reading



Beyond-worst case analysis

What is different?

Analyze algorithms in a more instance specific way

A Representative Problem

Intersect two sorted sets

Further Reading

http://timroughgarden.org/f14/f14.html

Algorithms for Data Science

What is different?

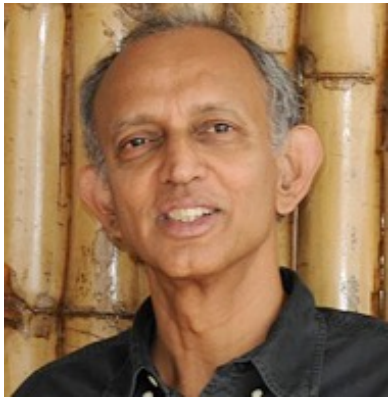
Algorithms for non-discrete inputs

A Representative Problem

Compute Eigenvalues

Further Reading





https://home.ttic.edu/~avrim/book.pdf

Algorithms and Society

What is different?

Measuring and correcting for harms caused by Algorithms

A Representative Problem

Bias in ML

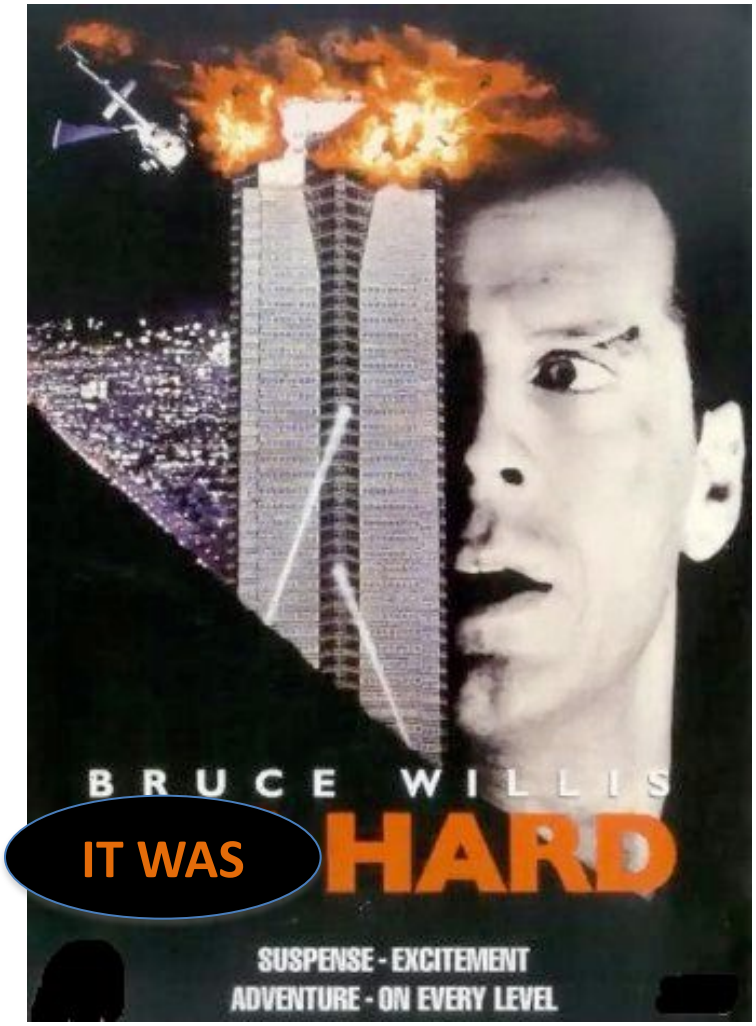
CSE 440 in Spring 24: ML and Society

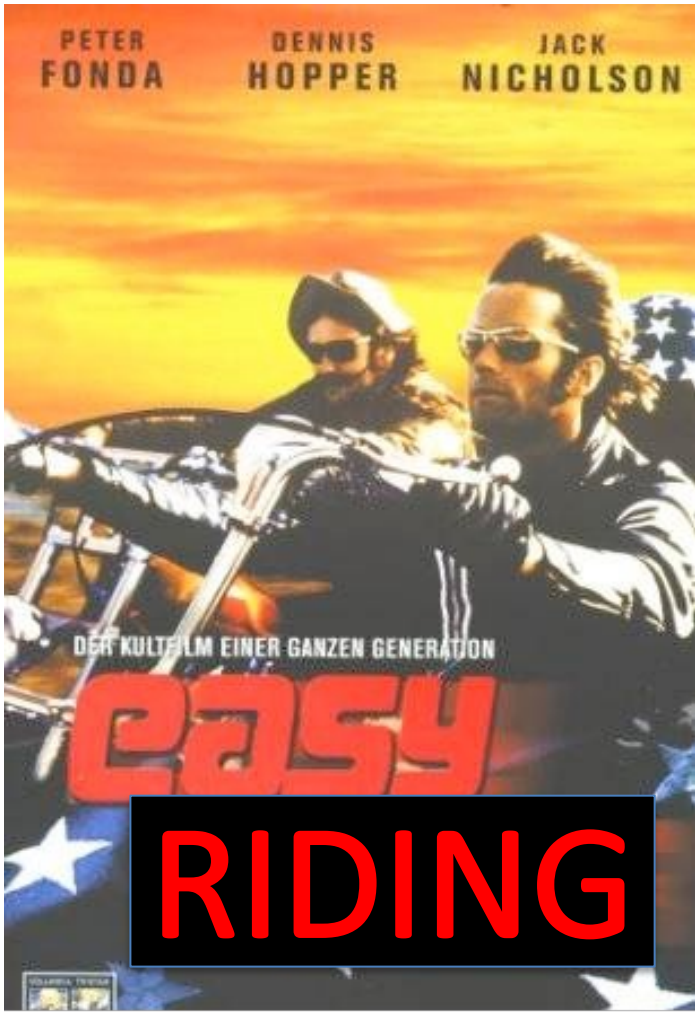
Q & A session

CSE 331 related Questions?

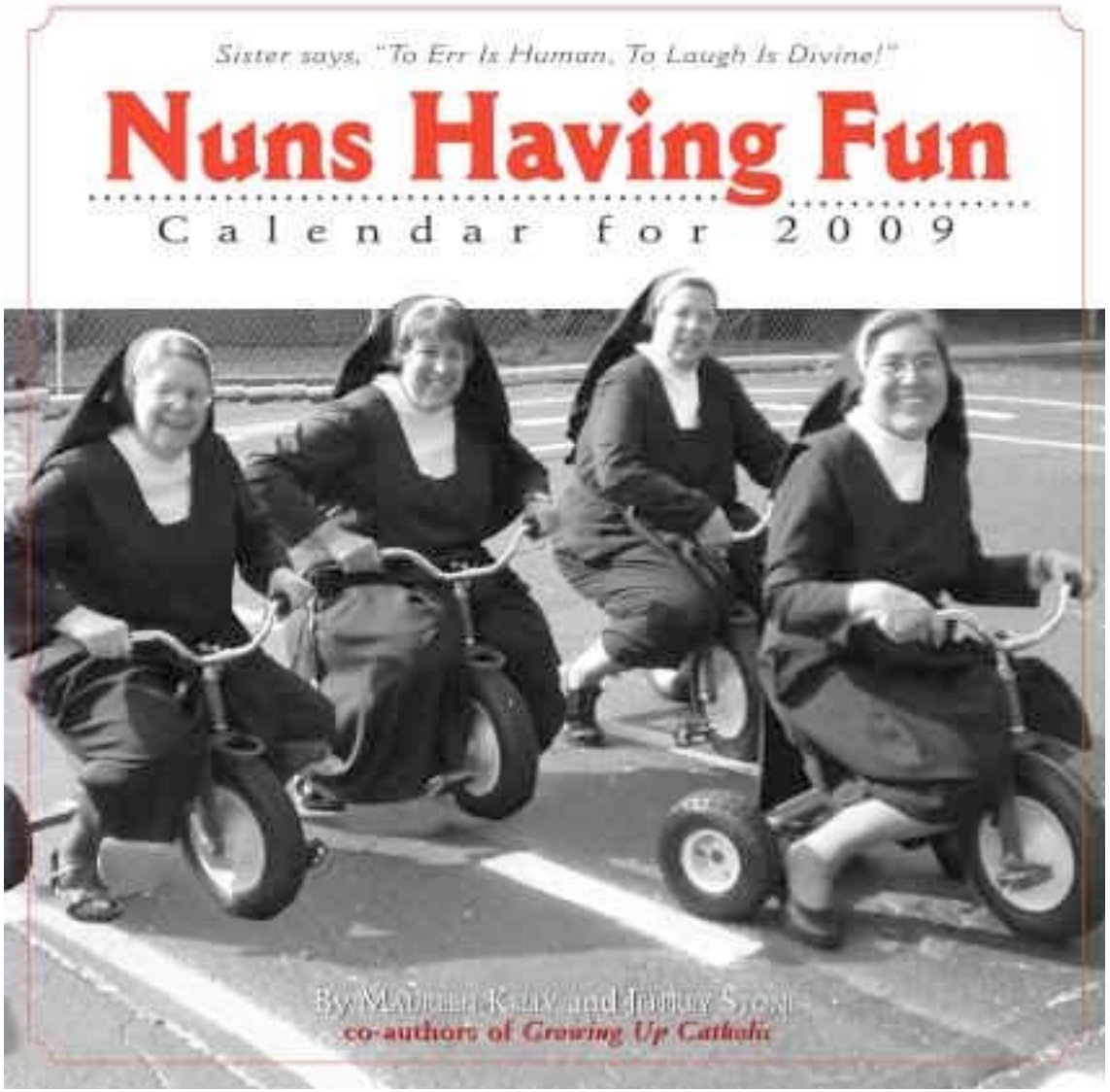
Any other Questions?

Whatever your impression of the 331





Hopefully it was fun!



Thanks!

Except of course Reflections 4+5, survey and the final exam