Lecture 39

CSE 331

Dec 7, 2016

Final exam piazza post

note ☆ stop following 41 views Actions

Final exam post

I'll start off with some generic comments:

- The final exam will be based on all the material we have seen in class till the lecture on Monday Dec 5 (i.e. up to the P vs NP stuff).
- The lecture on Wednesday, Dec 6 will be a Q & A session (where you can ask any 331 related questions)-- see @829.
- Exam will be from **noon to 2:30** on Friday, **Dec 16** in class (**KNOX 110**). Note that the exam will be for 2.5 hours and *not 3 hours* as it says on HUB.

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

1. Take a look at the sample final (@735) 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 deduction 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 (@735).
2. Stay tuned for more information on extra OHs (during the exam week).
3. Attend the Q&A session (Wednesday, Dec 6) in class.
4. 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.)
5. For the T/F questions (i.e. the first two questions), anything that was covered in class in fair game. If you want to refresh your

Mini project video grading

Will be done tonight

If your group is selected for presentation you'll be notified by tonight

Algorithms for Data Science

What is different?

Algorithms for non-discrete inputs

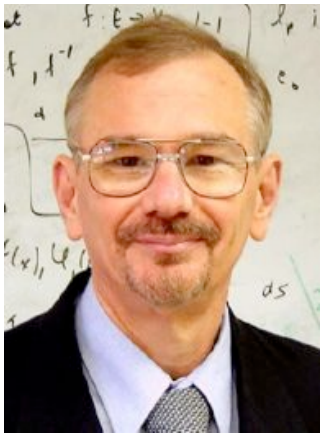
A Representative Problem

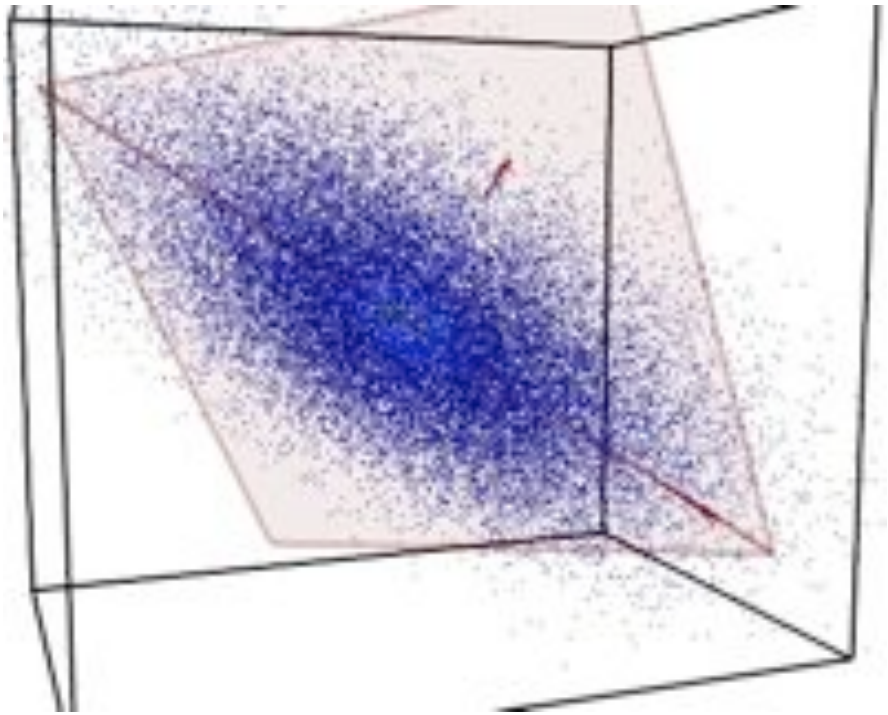
Compute Eigenvalues

Further Reading

http://research.microsoft.com/en-US/people/kannan/book-no-solutions-aug-21-2014.pdf

Johnson Lindenstrauss Lemma

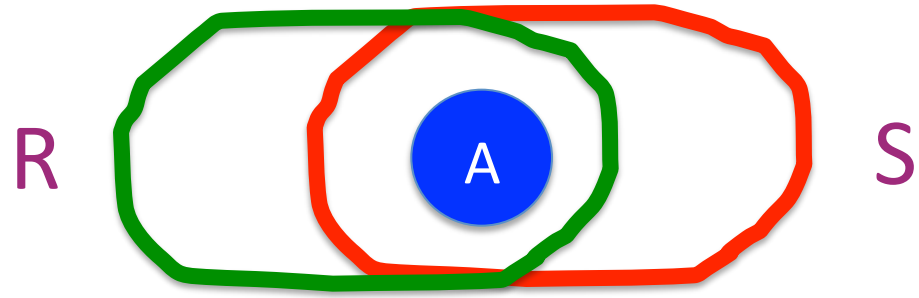




http://www.scipy-lectures.org/_images/pca_3d_axis.jpg

The simplest non-trivial join query

Intersection of R and S

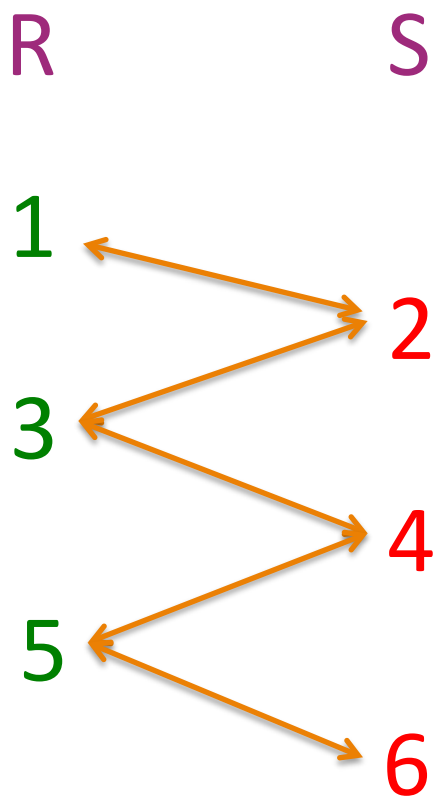


Assume R and S are sorted

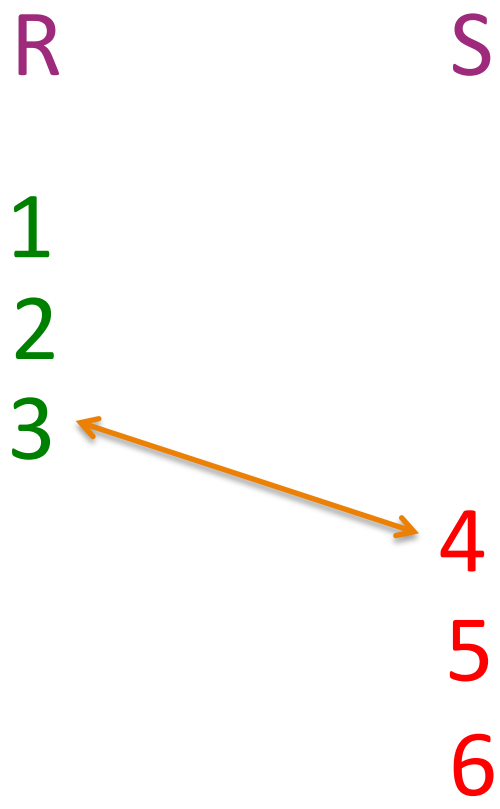
Let us concentrate on comparison based algorithms

Assume |R| = |S| = N

Not all inputs are created equal



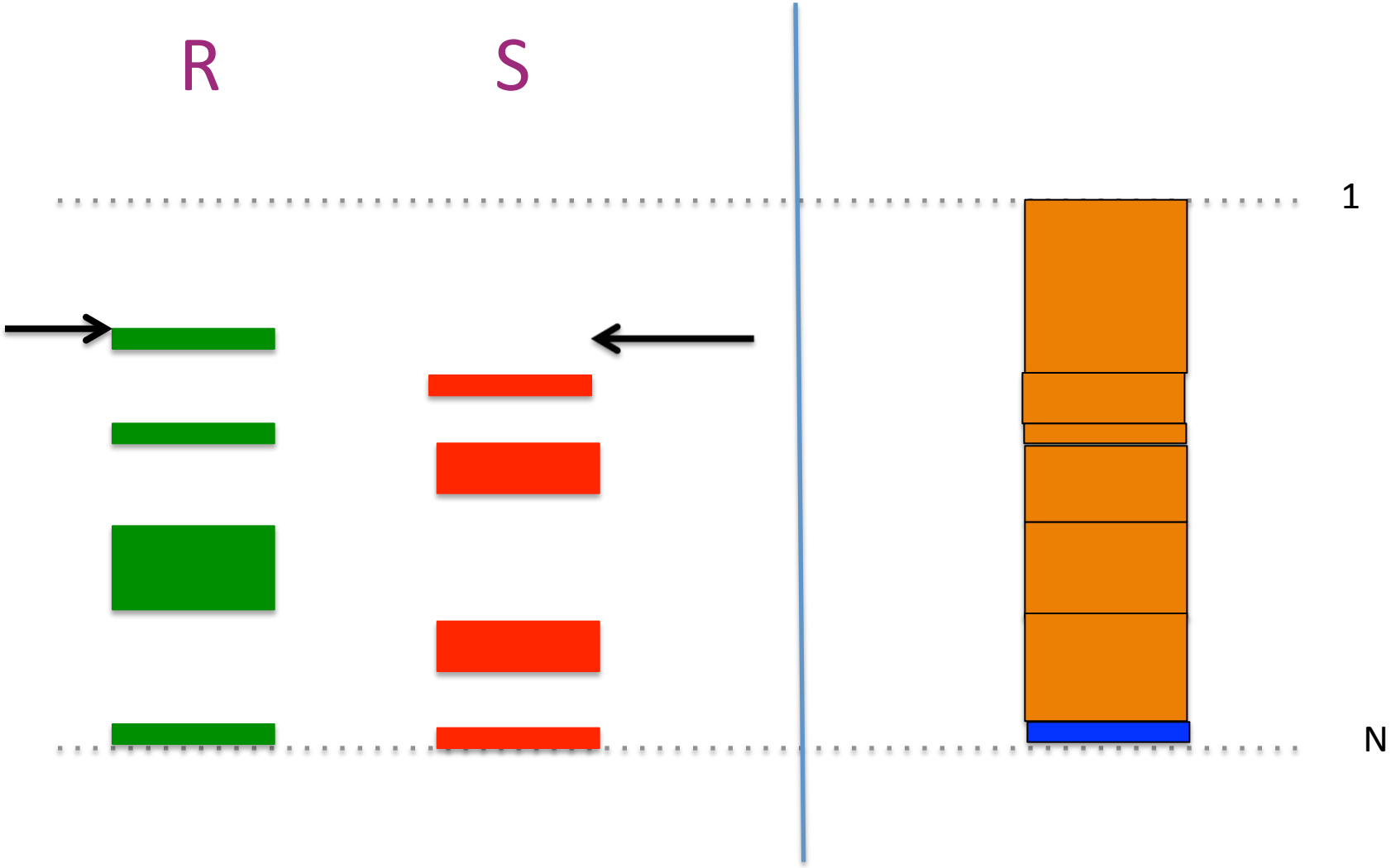
Ω(N) comparisons



1 comparison!

We need a faster/adaptive algorithm

The MERGE algorithm works



An assumption

Output of the join is empty

MERGE is (near) instance optimal

Benchmark: Minimum number of comparisons (C) to "certify" output

Demaine Lopez-Oritz Munro

C logN comparions (and time)

R S

1

Value not ruled out yet

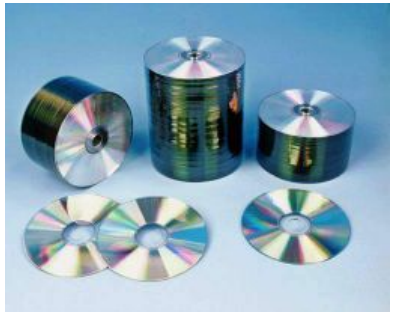
Need a comparison to rule the value out

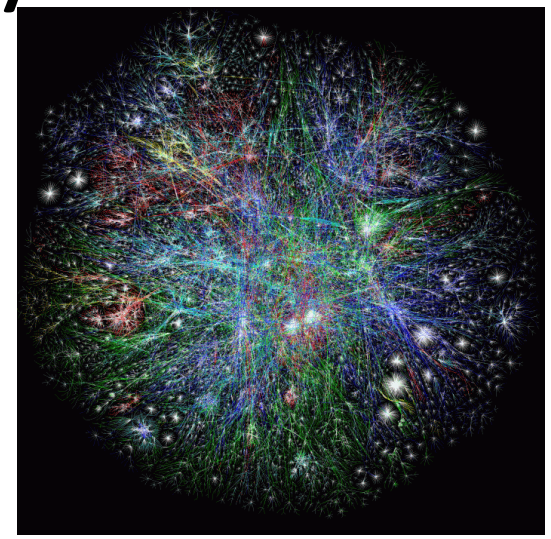
Each value involved with ≤2 comparisons

Once the pointer moves the value is never seen again

Each move takes log N comparisons

Coding Theory

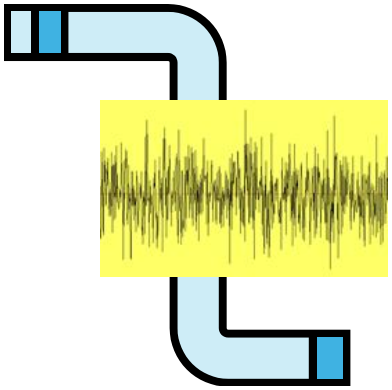




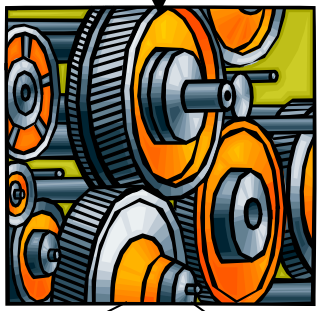
Communicating with my 5 year old

x

C(x)



y = C(x)+error



x

Give up

"Code" **C**

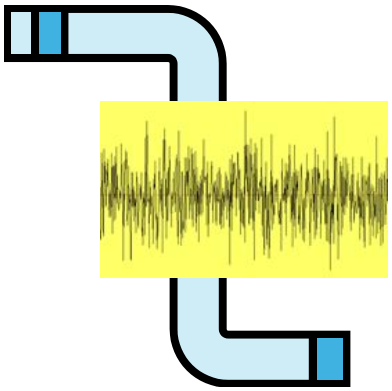
"Akash English"

C(x) is a "codeword"

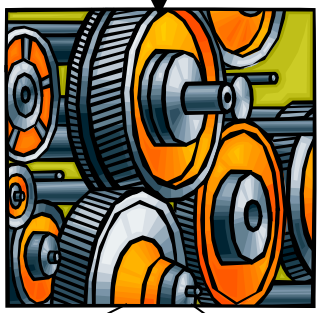
The setup

x

C(x)



y = C(x)+error



x

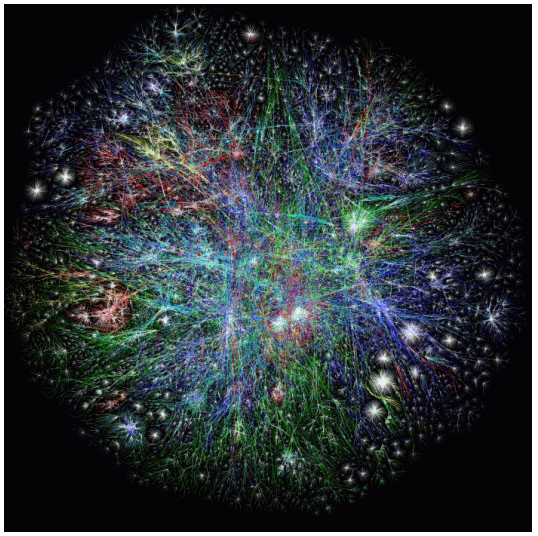
Give up

Mapping **C**

- Error-correcting code or just code
- Encoding: x → C(x)
- Decoding: y → x
- C(x) is a codeword

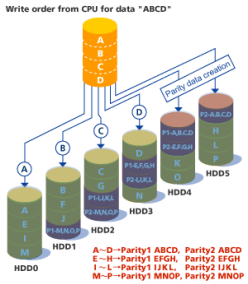
Different Channels and Codes

- Internet
  - Checksum used in mult layers of TCP/IP stack
- Cell phones
- Satellite broadcast
  - TV
- Deep space telecommunications
  - Mars Rover

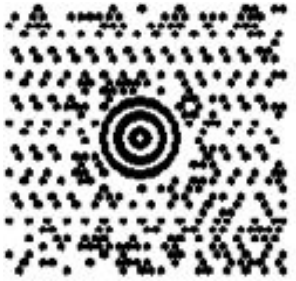


“Unusual” Channels

- Data Storage
  - CDs and DVDs
  - RAID
  - ECC memory



- Paper bar codes
  - UPS (MaxiCode)



Codes are all around us

Redundancy vs. Error-correction

- Repetition code: Repeat every bit say 100 times
  - Good error correcting properties
  - Too much redundancy
- Parity code: Add a parity bit
  - Minimum amount of redundancy
  - Bad error correcting properties
    - Two errors go completely undetected
- Neither of these codes are satisfactory

1 1 1 0 0	1

1 0 0 0 0	1

Two main challenges in coding theory

- Problem with parity example
  - Messages mapped to codewords which do not differ in many places
- Need to pick a lot of codewords that differ a lot from each other

- Efficient decoding
  - Naive algorithm: check received word with all codewords

The fundamental tradeoff

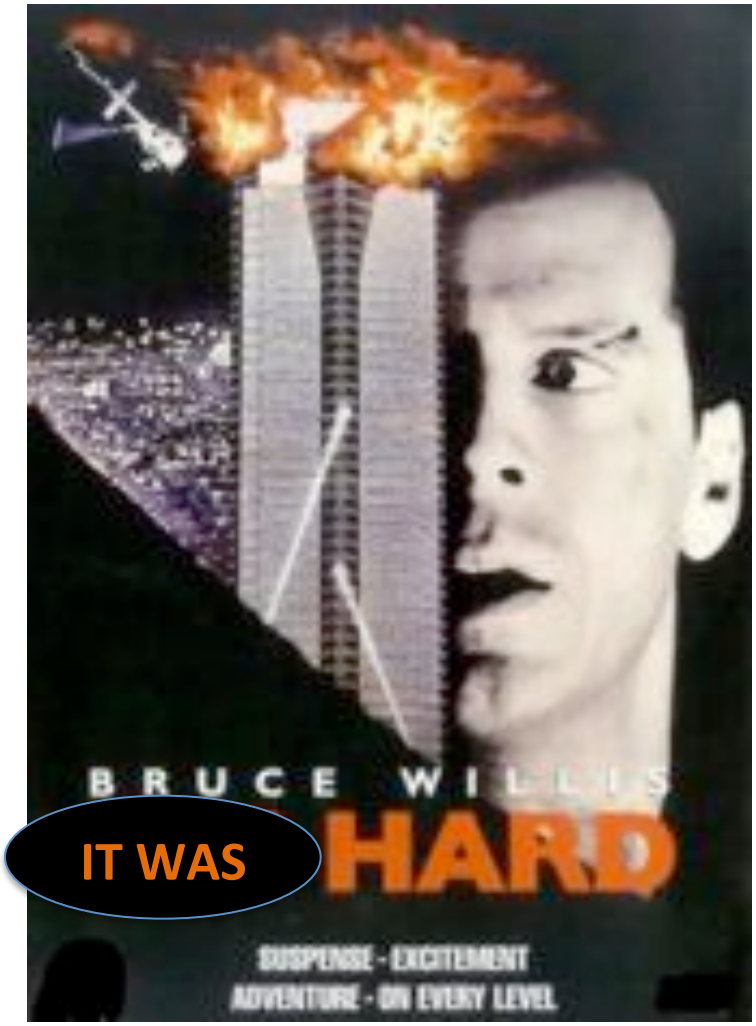
- Correct as many errors as possible with as little redundancy as possible

Can one achieve the "optimal" tradeoff with *efficient* encoding and decoding ?

Interested in more?

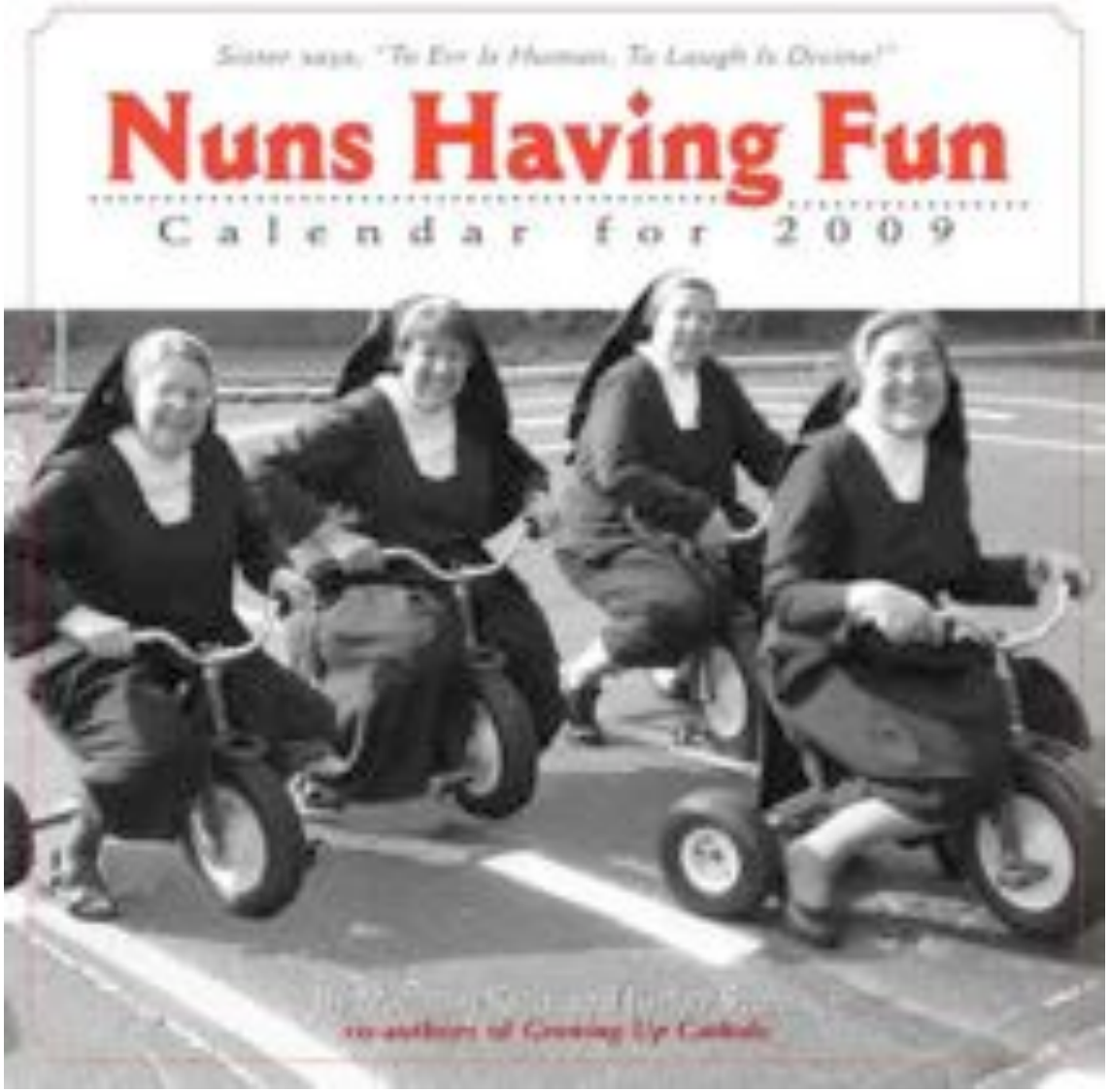
CSE 545, Spring 201?

Whatever your impression of the 331



IT WAS

Hopefully it was fun!



Thanks!

Except of course, HW 10, presentations and the final exam