Final lecture!
Final exam study tips
1. Aim for **beautiful**

Write full solutions (even when on paper!)
Practice checking your work

*Is there anything I’m not saying?*
*Is there a better way to do this?*
2. Plan out your studying; cover everything.

Space out your studying
Take breaks
Do a mock final

Make lists. *Definitions, concepts, techniques.*
3. The highest level of learning is creation.
4. Work with others.

share knowledge; teach each other
read over each other’s solutions
make up questions and trade

keep each other *focused and positive*
Hey folks,

Anybody want to study for the final together? I’m going to be in BA3200 on _____ at 1pm onwards, feel free to join! I’m looking to do really well on the final, so some serious studying will be happening. I’ll be the one wearing the AWESOME cat sweater.

P. S. This post is so ironic because I copied exactly what David showed us in class, lol.
5. Exam day

Sleep well. Eat well. Do something fun.
Arrive early (but not too early).
Bring a watch and a drink.

Skim every question. You have the time!
Play to your strengths.
Get key ideas down first, fill in details later.
Big ideas from CSC165

(aka “Things you should take away”)

Mathematical expression

Can I translate a statement into predicate logic?
Can I understand and use a definition?
Can I write unambiguous... English?
Mathematical reasoning

Can I write a formal proof header?
Can I write a formal chain of reasoning?
Can I understand and use an “external fact”? 
Can I use different proof techniques?
Algorithm analysis

Can I prove statements involving $O$/Omega/Theta?
Can I analyse the runtime of an algorithm exactly or by using approximations (at most/at least)?
Can I... worst-case/best-case/average-case ...?
Tackling new domains

Can I use definitions and prove statements about...
number theory
representations of numbers
graphs and trees
whatever else life throws at me
Looking ahead

(aka “Where will you use this again”)

“Theory” courses

236/240
263/265
373
“Theory” courses

304 (game theory)
336 (numerical computing)
343 (databases)
384 (artificial intelligence)
But really:

any time you need to think and communicate ideas and arguments precisely
Limits of Computation
Story so far

Here’s an algorithm.
How long does it take to run?
A “meta” question

Here’s a problem.
Is there an efficient algorithm to solve it?
Satisfiability

Here’s a propositional formula.

\[(p \text{ AND } q) \text{ OR } (\text{NOT } p \text{ AND } r)\]

Is there a truth assignment that makes it true?
The (literally) million-dollar question

Is there an efficient algorithm to solve satisfiability?
Another “meta” question

Here’s a problem.
Is there an efficient algorithm to solve it?
The Halting Problem

Given a program \texttt{my\_prog} and value \texttt{x}, determine whether \texttt{my\_prog(x)} halts.
def halts(my_prog, x):
    """Return True if my_prog(x) halts, and False otherwise. """
The end.