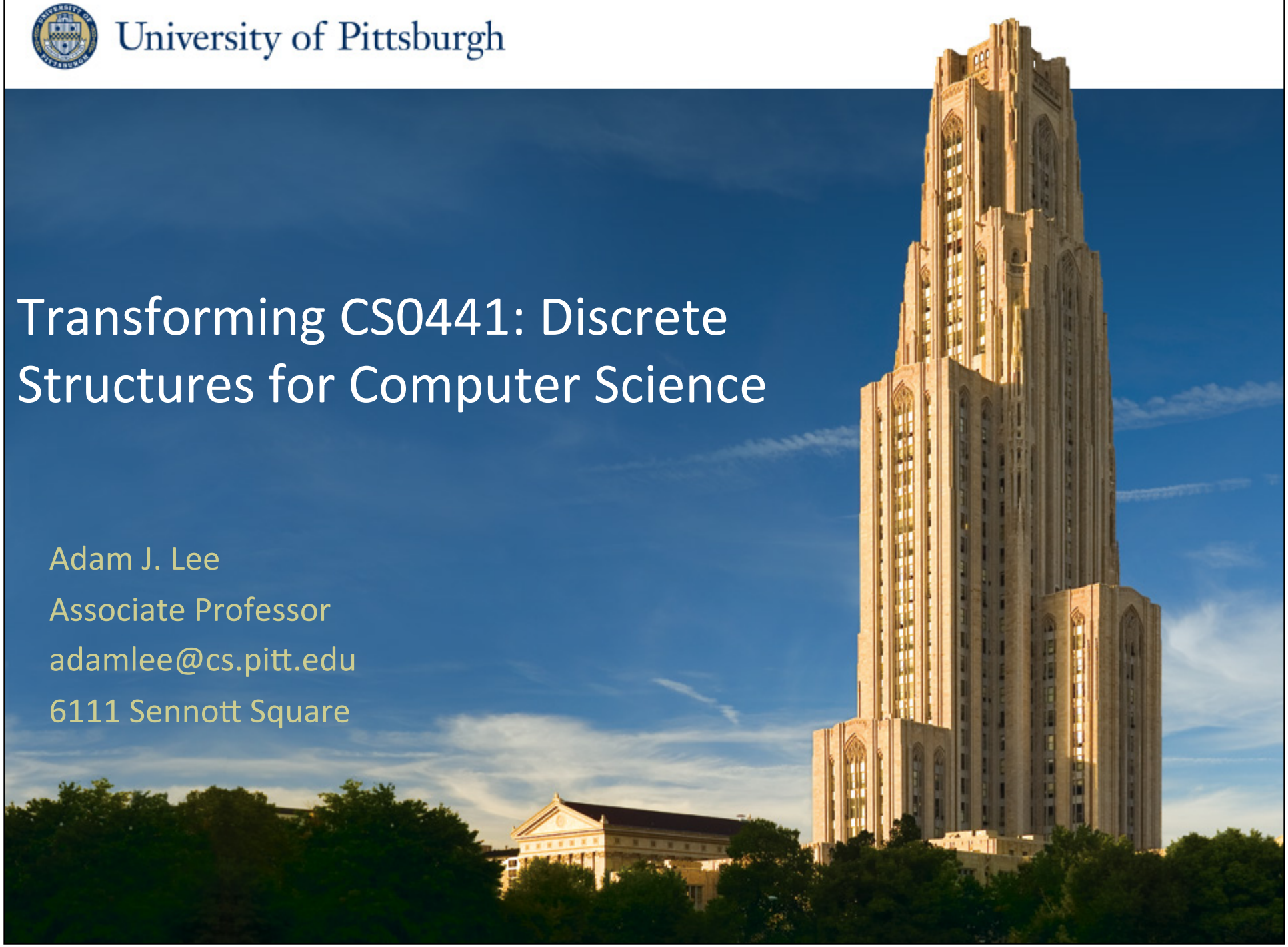


Transforming CS0441: Discrete Structures for Computer Science

Adam J. Lee
Associate Professor
adamlee@cs.pitt.edu
6111 Sennott Square

Roadmap

- About the team
- What is CS 0441?
- What problems are we hoping to address?
- How are we planning to do this?
- Current state of affairs

This transformation is a team effort

Faculty:

- Adam J. Lee, Associate Professor
- Thumrongsak Kosiyatrakul (Tan), Lecturer
- Daniel Mosse, Professor

Undergraduates:

- Dustin Martin
- Nick Treu
- *TBD*

What is CS 0441?

One of two entry points into the CS program

- CS 401 -> Learn to program
- CS 441 -> Learn to reason

From the course catalog: *The purpose of this course is to understand and use (abstract) discrete structures that are backbones of computer science. In particular, this class is meant to introduce logic, proofs, sets, relations, functions, counting, and probability, with an emphasis on applications in computer science.*

What is discrete mathematics?

Discrete mathematics is the study of *distinct* objects or structures and their relationships to one another

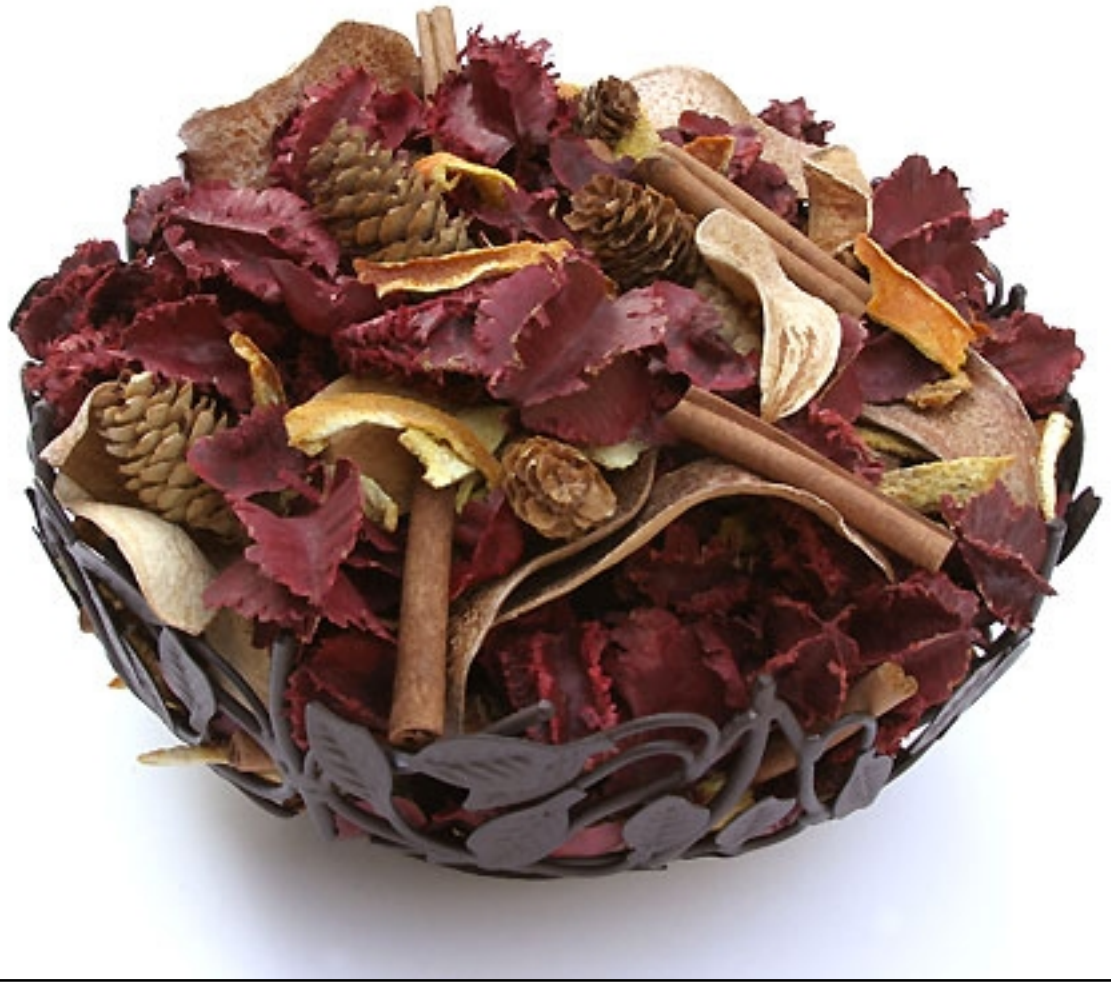
For example:

- How many ways can a valid password be chosen?
- Can traffic flow between two computers in a network?
- How can we transform messages to hide their contents?
- How do we parse a given sequence of commands?

Contrast: continuous mathematics (e.g., calculus) which study objects and relationships that vary continuously

- E.g., Position, velocity, and acceleration of a projectile

What is CS 0441, *really*?



Students (across universities) have mixed opinions about discrete mathematics

Shouldn't we be writing code? What does these topics actually have to do with computers?

My opinions:

- Computer Scientist =/= Programmer
- Diversity of discrete mathematics topics -> huge future impact

Trying to break preconceptions about discrete math:

- Each lecture ties topics back to future courses, recent events, etc
  - Logic & proofs -> AI & security; Combinatorics -> Networks & Security; Relations -> Databases; etc…
  - Lots of hands-on exercises in class
  - Interactive lecture style

So, Adam, why are you transforming 441?

Important: Over the last few years, our enrollments have doubled. Our faculty has stayed the same size.

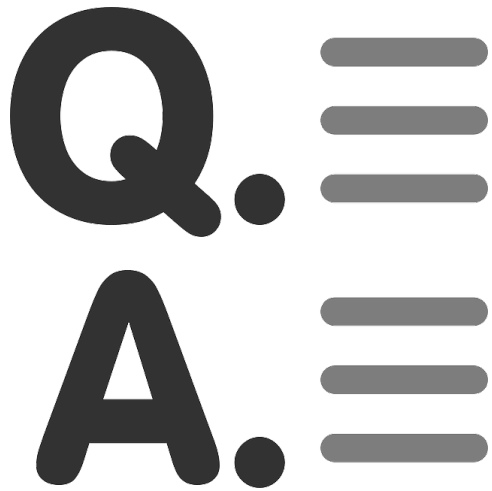
Now, I'm a computer scientist. I *really* care about scalability. This breaks it.

A few examples:

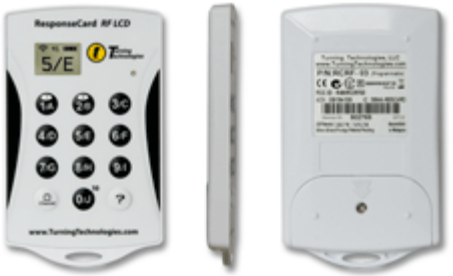
- Managing in-class problem solving breaks
- Supporting interactive lecture and proof writing
- Micro-assessment of individual performance
- …

We are transforming the course in three ways

Flipping certain topics



Self-assessment database



Clickers for guided work

Why flipping? Which topics?

Not all topics are created equal. Historic toughies:

- Logic programming and nested quantifiers
- Mathematical induction*
- Permutations and combinations
- Expected value and variance

Using self-assessment DB!

Our plan will be based upon (an optimized version of) my recent strategy for teaching expected value and variance

- Before class: ~20 minute video hosted on My Pitt Video
- Before class: Short quiz hosted on course page
  - ➢ Gauge understanding, sort out areas of confusion
- In-class: Problem-solving session
  - ➢ Examples on the board, small group work time, etc

Goal: Quicker exposure to tricky areas. More opportunity for nuanced explanation in the classroom.

Clickers!

How can we take the pulse of the classroom?

My existing method for teaching relies heavily on feeling out the class

- Circulating and sitting in with groups during exercise breaks
- Talk to students by name before, during, and after class

Size & room constraints will make the first of these tough in a large format setting, genetic limitations may hinder the second

Plan: Use clicker questions for all in-class exercise break problems

- Each choice stems from a common type of mistake*
- Response breakdown from the class -> sources of confusion (?)

Goal: Be more adaptive in how I go over these problems with students

- How much time is spent on re-working/explaining the problem?
- Which common mistakes are worth highlighting?

We are developing a (self-)assessment database for use by instructors and students

Observation: Clicker-format questions are useful for *explaining* incorrect answers and *identifying* areas for practice.

Why not expose more of these questions to students?

Design goals:

- Allow students to generate random problem sets by topic area
- Allow recitation TAs to generate in-class quizzes
- Allow intructors to generate out-of-class assessment quizzes
- Provide explanations for errors, additional study resources

Flipped topics!

Currently under development

- So far, most time spend under-the-hood
- Basic prototype should be functional later this week

Sample question

A particular brand of shirt comes in 12 colors, has a male and female version, and has three sizes for each sex. How many unique types of this shirt are available?

A. 72 ✔

B. 17

C. 18

D. 27

Incorrect. This question is an application of the product rule: There are three *independent* choices (i.e., color, sex, size). Given 12 choices for color, 2 choices for sex, and 3 choices for size, there are 12x3x4=72 ways to pick a shirt.

Note: Did you use the *sum* rule instead of the *product* rule? The sum rule should only be used when choices are mutually-exclusive.

If you'd like more practice on these types of problems, please see problems 3,5,7,9,11, and 13 in Section 6.1 of the textbook.

How will we evaluate our transformation?

We will evaluate the efficacy of our transformation experimentally

- Offering 1: MW 11-12:15 (Max 48 students)
- Offering 2: TH 11-12:15 (Max 125 students)

Both taught by me

Beginning of the semester:

- Assessment of both populations via short exam
- Conducted by Tan, who is on our undergraduate assessment committee

During the semester:

- Offering 1: Taught in the manner that I historically teach CS 0441
- Offering 2: Transformed class
- Joint midterm and final exams; useful for comparative analysis

After the semester:

- Assessment of outcomes
- Possible confounding factor: This will be the largest class I've taught…

What's the current state of things?

Things are currently moving along:

- Curriculum: Slight flux due to restructuring theory sequence
- Self-assessment: Database built, web UI in progress
- Clicker questions: Being put together as part of DB effort
- Flipped topics: Identified, short lectures are being planned

Still to be done this semester:

- Self-assessment: Complete working prototype, fully populated question bank
- Clicker questions: Completed and incorporated into lecture notes
- Flipped topics: Videos recorded

Next semester: Run our experiment, keep everything rolling

Plans to hire question writers (in process) and retain Dustin as self-assessment admin.

Thanks!

(More) Questions? Comments?