• Diverse interests and background!
• Short “lightning” talks today
  • Should follow-up individually for more info!
Summer Research in CS

- Spend 8-10 weeks working with individual faculty member on specific research problems
- Live in dorm with other students (not just CS students)
- Work in computer lab (mostly)
- “Flexible” schedule
- Social events (epic canoe trip)
- Chance to get to know us personally
- No classes = more time for faculty to focus on research
Honors Thesis in CS

- Spend senior year working on independent research project
- Advised by faculty member
- Fall is spent reading
  - Become an expert in chosen area
- WSP is spent working...hard
- Spring is spent writing
  - Gathering results
  - Writing thesis
- Present results in May in public presentation
- Lots of work, but very rewarding
- Highly recommended for those considering grad school
Faculty research interests
Jeannie Albrecht

- Jeannie works in Distributed System, Networks, IoT
- Lately, Smart Buildings
- Also K-8 CS ed
Traditional Energy Use Visualizations

- Need something that is more intuitive...
Research Goal: Make Buildings *and* Occupants Smart(er)

- Use *ambient* visualizations to help occupants better understand energy usage (in kitchen)
- Research students need to have at least taken CS 136
Welcome to the Kitchen Simulator!

Breakfast is ready! Leave this page open while you finish the survey.
Code Word: monkey
Duane Bailey

- Duane is a computer architect
- Works with hardware
- Often advises projects that use FPGAs
- FPGAs contain programmable logic blocks and interconnection circuits
- They can be programmed/ reprogrammed to the required functionality after manufacturing
- (kinda like custom built hardware)
- Need at least CSCI 237
Dan Barowy: Spreadsheet Support for End-to-End Data Analysis

Data wrangling
FlashRelate
PLDI '15

Data cleaning
CheckCell
OOPSLA '14

Debugging
ExceLint
OOPSLA '18

Background req’d: 136!!! 334 or 376 preferred, but not required.

Daniel Barowy, Assistant Professor
Andrea Danyluk

- Andrea works in Machine Learning
- Projects involve “Supervised Inductive Machine Learning”
- Examples:
  - Learn a predictive model from labeled training data
    - Will this cancer patient experience a recurrence after initial treatment? Is this banknote authentic or not? Who is the author of this Amazon review?
  - Recent excitement
    - Availability of data. Increased processing power.
- Interesting research problems
  - New algorithms for new types of problems. Model interpretability for humans. Models that balance accuracy and cost. And many more.

https://www.geeksforgeeks.org/machine-learning/
Supervised Inductive Machine Learning: Summer Research at Williams with Andrea

Application-Driven

Model expressivity

Identify individuals

Identify keys to successful teaching and learning

What it’s like:

• Learn application domain
• Learn ML algorithms and empirical methodology
• Gather, clean, and featurize data (at least CSCI 136)
• Run experiments
• Have fun!
Steve Freund

- Steve works in Programming Languages
- Probably need to have taken at least 136 and probably 334
* Bug fixes and performance improvements
Amazon Prime Video
Jan 29, 2018

Bug fixes and performance improvements
* Bug fixes and performance improvements
Bug fixes
HBO GO: Stream with TV Package

3 Days Ago

- Bug fixes and performance improvements.
General bug fixes and performance improvements
Google Calendar

Today

- Bug fixes and performance improvements.
Program Specification

Slow and Buggy Code

Traditional Software Process
With Better Programming Tools

Better:
• Programming Languages
• Verification Techniques
• Debuggers
• Programming Environments
• ...
Work on Programming Languages And Analysis Tools

• language design
• theoretical foundations
• proving theorems
• systems development
• performance modeling
• experimental validation
Hi folks! My name’s Iris Howley. I’m currently at the AAAI Conference on Artificial Intelligence, Ethics and Society with a couple of my Williams research students.

There’s more info on doing research with me on my website under “Info for Students” so check that out & contact me to talk if you’re interested.

MIT Tech Review recently wrote up some related work on my topic, here:

www.bit.ly/AIexplain

Iris Howley, iris@cs.williams.edu
Assistant Professor of Computer Science
Iris works at the intersection of Human-Computer Interaction, Artificial Intelligence, and Education. Here are some questions she thinks about!

Questions:

- How do users of ed tech systems understand the algorithms underlying their systems?
- Does increased algorithmic understanding lead to more informed decision-making?

Skills We Learn in this Research:

- **Understand humans**: Interviews, User studies
- **Build systems to explain AI**: Javascript, web dev, computer languages
- **Evaluate**: Survey measures, behavioral measures
Iterative User-Centered Design

And now for some example of past summer research student work from my group!
Iterative User-Centered Design

Unusual slip-up:

The person knows the skill, but something unusual happened, such as a mind-blank or a small error, that caused them to answer incorrectly.
Iterative User-Centered Design

We associate the initial level of mastery with this vial to begin with.

Something unusual happened, an error, that caused them to answer.

They guessed, continuing straight. The journey will take even longer now.

In BKT, a student answers incorrectly, this may be either a Slip or a Guess. A Slip predicts that the student knows the information, but made a mistake answering the question. A Guess predicts that the student does not know the information.

Left, left, right, left... They are on foot way again. The destination is approaching and it looks like the driver does know how to get there after all.

In BKT, many correct answers increase the odds that the student knows the skill, until eventually, a certain cutoff is reached, and the algorithm determines that the skill has been learned.

They've arrived!

The trip took a little longer than expected, with the accident Slip and the Guess the driver made, but in the end, they're happy they know how to get to their destination. Their Prior Knowledge of the route set them up for a good start, and every Correct Choice they made helped them overcome their small mistakes.
We associate the initial level of mastery with this vial to begin with.
Iterative User-Centered Design

We associate the initial level of mastery with this vial to begin with.

BKT BALLOON SIMULATOR

Drag/click the sliders on the right to adjust the parameters and help the balloon rise. Explore on your own or use the following prompts as a guide.

1. Find two different parameter combinations that will result in mastery if the student answers correctly. Hint: make P(learned if correct) > 0.95 and press "answer correct" to verify your results.

2. What happens to P(learned if correct) and P(learned if wrong) if P(guess) and/or P(slip) exceeds 0.5?

3. What happens to P(learned) if the student answers incorrectly? Hint: compare P(learned if wrong) with P(known) (aka, your previous P(learned)).

4. Keep exploring! Can you find any other flaws or interesting characteristics of BKT?

Additional Parameters

Drag the sliders below to adjust the values of the parameters with which P(known) is calculated. Play the memory game again. Try to obtain mastery for different slider combinations.

- P(learning): 0.2
- P(slip): 0.05
- P(guess): 0.14

**The default value of P(guess) is \( \frac{1}{2} \), the probability of a random match.**
Fun

Stickers coordinating with our research posters

Making stickers to match our research posters!
Bill Jannen

Bill works in Storage Systems
Data Structures
Storage Research Goals

Bill tries to build systems that help users:

- Read and write their data faster (performance)
- Store more data in the same physical device (capacity)
- Ensure that data is never lost (safety)

He’s interested in topics like:

- New HDD and SSD technologies
- File systems
- Key-value stores

- Data deduplication
- Write optimized data structures
- Filters

His research group has a website with their project info: www.betrfs.org

Probably need CSCI 136 and 237 (ideally)
How Bill Spends His Summer Vacations
How To Draw a Graph
A (2,2)-Planar Drawing
The Game

• Identify interesting/useful/important class $\mathbf{C}$ of graphs

• Specify drawing model $\mathbf{M}$

• Ask questions
  • Can $G \in \mathbf{C}$ be drawn according to model $\mathbf{M}$?
  • Can the drawing be computed efficiently?
  • Can the class $\mathbf{C}_\mathbf{M}$ be easily characterized?

• Need CSCI 256 and strong background in math
Sam McCauley

- Sam works on Data Structures
- How do we store large amounts of data so that we can quickly and efficiently access it?
Data Structures (Sam McCauley’s research)

• Main principle: use randomness (and some math) to make these data structures simpler and more efficient

• Combines coding skills with worst-case analysis

\[ E \left[ \sum_{i=1}^{k_i} \left| \{q \in Q_0 \mid h_\beta(q) = h_{\beta'}(x_i) \text{ for some } \beta, \beta' \in \{0, 1\}\} \right| \left| q_i \text{ is costly} \right| \right] \leq 2 + 2k_i|Q| (\epsilon/n) \]

Using probability (like this) can lead to improved performance!
What’s needed?

• Flexible—both coding and math skills are useful
• But, you’ll probably want to have taken Algorithms (CSCI 256)

• Projects may entail:
  • Designing new data structures
  • Implementing data structures that work in theory (how well do they do in practice?)
  • Looking at what use cases work best
Kelly Shaw

- Kelly works on Parallel Architectures/IoT
- Interest in systems running concurrent applications
- Evaluation of existing systems
  - Analysis of application needs
  - Analysis of application/platform correctness
- System/tool creation
  - Static analysis tools examining correctness of applications
    - Amazon AWS IoT, Samsung SmartThings, Google IoT
  - Modifying existing systems to try different solutions related to concurrency
    - Mozilla WebThings – gateway for smart home devices
    - Cassandra – distributed database
- Prerequisite knowledge: CS237
Name. **Shikha Singh**

Research Area. **Algorithmic Game Theory**

- Problems at the intersection of economics and computation
- Overarching goals of the field:
  - Design the rules of the game (a.k.a algorithms) so that strategic behavior by participants leads to a desirable outcome
  - Understand and analyze selfish behavior

*The next time we bemoan people exploiting loopholes to subvert the intent of the rule makers, instead of asking ‘What’s wrong with these people?’ let’s instead ask, ‘What’s wrong with the rules?’ and then adopt a scientifically principled approach to fixing them* — Hartline and Kleinberg
Algorithmic Game Theory: Examples of Problems

• **Auctions.** Designing revenue maximizing auctions which incentivize bidders to bid their true value

• **Resource allocation.** Designing a fair allocation algorithm that incentivizes selfish participants to reveal true preferences

• **Selfish routing.** Analyzing global congestion in networks where each participant selfishly minimize their travel time
Algorithmic Game Theory: Background

• What do you need to do work on work in this area?
  • Mathematical background, such as Discrete Mathematics
  • Some familiarity with designing and analyzing algorithms
  • Programming background useful but not necessary
Other Research: Algorithms and Data Structures

- I/O-efficient algorithms and data structures
- Randomized and adaptive data structures
- Optimization, Approximation, online and LP duality algorithms
Aaron Williams

Retrogame Archeology / History
How were video games from the 1980’s programmed?
We examine the machine code and figure it out!

This is a new field and there are a lot of “important” problems to be solved.

+ Play study video games all summer!
+ Low-level system design and programming.
+ Learn undocumented programming tricks and optimizations from masterful coders of yore.

Requirements: Programming experience.
Computational Complexity of Puzzles and Games

Which type of puzzle is harder: Sudoku or Rush Hour?

Apply the area of the computational complexity to well-known puzzles and games.

+ Creative thinking.
+ Experience + supervision writing a research paper.
+ Turn computational complexity into an applicable skill for use later in your career.

Requirements: CSCI 256 (required) / CSCI 361 (preferred).
import itertools

for permutation in itertools.permutations([1, 2, 3, 4, 5]):
    # do something with the permutation

How is the above code implemented?
Why are the Python developers doing it wrong?

+ Discrete Mathematics and Algorithms.
+ Experience + supervision writing a research paper.
+ Implement practical libraries that are used in high-efficiency applications.

Check out combos.org for more details.

Requirements: CSCI 256 / MATH 200
Next Steps

• Come talk to us!
• Summer RA apps due Feb 17
• Thesis apps due Apr 10
• Forms can be found on dept website under research