

REU-CAAR: Research Experience for Undergraduates in Combinatorial Algorithms and Applied Research

Andrew Childs
John Dickerson
Soheil Feizi
Furong Huang
William Gasarch
Thomas Goldstein
Jon Katz
Josh Twitty

Time and Money

1. June 8- August 14 (10 weeks) (Actually arrive on June 7)
2. \$6000 stipend
3. Free housing
4. ~ \$300 meal money
5. Excellent student-to-teacher ratio.
6. For more info
<http://www.cs.umd.edu/projects/reucaar/index.html>

Activities

1. Research

Activities

1. Research
2. Learning more discrete math and algorithms.

Activities

1. Research
2. Learning more discrete math and algorithms.
3. Research

Activities

1. Research
2. Learning more discrete math and algorithms.
3. Research
4. Lunches

Activities

1. Research
2. Learning more discrete math and algorithms.
3. Research
4. Lunches
5. Research

Activities

1. Research
2. Learning more discrete math and algorithms.
3. Research
4. Lunches
5. Research
6. Field Trip to the International Spy Museum

Activities

1. Research
2. Learning more discrete math and algorithms.
3. Research
4. Lunches
5. Research
6. Field Trip to the International Spy Museum
7. Research

Activities

1. Research
2. Learning more discrete math and algorithms.
3. Research
4. Lunches
5. Research
6. Field Trip to the International Spy Museum
7. Research
8. Presentation on How to do Bad Science.

Activities

1. Research
2. Learning more discrete math and algorithms.
3. Research
4. Lunches
5. Research
6. Field Trip to the International Spy Museum
7. Research
8. Presentation on How to do Bad Science.
9. Research

Activities

1. Research
2. Learning more discrete math and algorithms.
3. Research
4. Lunches
5. Research
6. Field Trip to the International Spy Museum
7. Research
8. Presentation on How to do Bad Science.
9. Research
10. Making friends for life! (now possibly with Facebook).
(WOW- When I first ran the program this was a novel idea!)

The Research

1. At the website is an application form.
2. When you apply you choose which projects you will be happy to work on, and rank them.
3. Next slides describe **THE PROJECTS!**
4. The projects range from THEORY to PRACTICAL WITH A THEORY BEND.

Sorting by Reversals (Andrew Childs)

Sorting by Reversals (Andrew Childs)

1. **Sorting** Can do with $O(n \log n)$ comparisons and cannot do better.
2. **Reversals instead of Comparisons** What if the following operation can be done in 1 step: Take the elements x_i, x_{i+1}, \dots, x_j and replace it with x_j, x_{j-1}, \dots, x_i .
3. **Many Questions** Given a list, how many reversals do you need to sort (NP-hard)? What if we can do the operation in parallel on disjoint parts of the list?

Sorting by Reversals (Andrew Childs)

1. **Sorting** Can do with $O(n \log n)$ comparisons and cannot do better.
2. **Reversals instead of Comparisons** What if the following operation can be done in 1 step: Take the elements x_i, x_{i+1}, \dots, x_j and replace it with x_j, x_{j-1}, \dots, x_i .
3. **Many Questions** Given a list, how many reversals do you need to sort (NP-hard)? What if we can do the operation in parallel on disjoint parts of the list?

Project Get upper and lower bounds on the question of parallel sorting with reversals as the basic operation.

Prereq Mathematical Maturity

Diversity in Matching Markets (John Dickerson)

Diversity in Matching Markets (John Dickerson)

Machine Learning: A Double Edges Sword:

- ▶ A firm gets lots of applicants for a job. Who to choose?
- ▶ The number of parameters is huge!
- ▶ How to winnow it down to just a few candidates?
- ▶ Good News— You can train an ML to do the job for you!
- ▶ Bad News— This may reinforce prior bias and make them baked in.

Diversity in Matching Markets (John Dickerson)

Machine Learning: A Double Edges Sword:

- ▶ A firm gets lots of applicants for a job. Who to choose?
- ▶ The number of parameters is huge!
- ▶ How to winnow it down to just a few candidates?
- ▶ Good News— You can train an ML to do the job for you!
- ▶ Bad News— This may reinforce prior bias and make them baked in.

Project How to use ML to evaluate without bias.

Prereq Basic understanding of Supervised ML, some programming.

Using Generative Adversarial Networks to Guarantee Fairness (Soheil Feizi)

Using Generative Adversarial Networks to Guarantee Fairness(Soheil Feizi)

Machine Learning: Need to try to fool an ML.

Using Generative Adversarial Networks to Guarantee Fairness(Soheil Feizi)

Machine Learning: Need to try to fool an ML.

I want an ML to recognize a dog.

Using Generative Adversarial Networks to Guarantee Fairness(Soheil Feizi)

Machine Learning: Need to try to fool an ML.

I want an ML to recognize a dog.

- ▶ We need lots of pictures of dogs. Thats easy.
- ▶ We need lots of pictures of things that are not dogs but could challenge the program. Thats hard.

Using Generative Adversarial Networks to Guarantee Fairness(Soheil Feizi)

Machine Learning: Need to try to fool an ML.

I want an ML to recognize a dog.

- ▶ We need lots of pictures of dogs. Thats easy.
- ▶ We need lots of pictures of things that are not dogs but could challenge the program. Thats hard.

Project Use a **Generative Adversarial Network (GAN)** to generate images that will challenge an ML. Also build an ML. Work both sides of this.

Prereq Probability, Linear Algebra, Python Program Preferred but not required.

Machine Learning For Ramsey Games and Nim With Cash (William Gasarch and Joshua Twitty)

Machine Learning For Ramsey Games and Nim With Cash (William Gasarch and Joshua Twitty)

Here is a game.

1. Board is $\{1, \dots, 27\}$. Players are **R** and **B**.
2. Players alternate with **R** going first.
3. On a player's turn he colors a number. **R** colors a number **RED**, **B** colors a number **BLUE**.
4. The first player to have an equally spaced seq of length 4 in their color wins.

Machine Learning For Ramsey Games and Nim With Cash (William Gasarch and Joshua Twitty)

Here is a game.

1. Board is $\{1, \dots, 27\}$. Players are **R** and **B**.
2. Players alternate with **R** going first.
3. On a player's turn he colors a number. **R** colors a number **RED**, **B** colors a number **BLUE**.
4. The first player to have an equally spaced seq of length 4 in their color wins.

Math R has a winning strategy. Proof does not give strategy.

Math-people have said that it would be very hard to find one.

CS With AI techniques from ML we can train a program to do well.

Machine Learning For Ramsey Games and Nim With Cash (William Gasarch and Joshua Twitty)

Here is a game.

1. Board is $\{1, \dots, 27\}$. Players are **R** and **B**.
2. Players alternate with **R** going first.
3. On a player's turn he colors a number. **R** colors a number **RED**, **B** colors a number **BLUE**.
4. The first player to have an equally spaced seq of length 4 in their color wins.

Math R has a winning strategy. Proof does not give strategy.

Math-people have said that it would be very hard to find one.

CS With AI techniques from ML we can train a program to do well.

Project Train a program to play games like these very well.

Prereq Linear Algebra, Prob, Stat. ML not required.

Generating Information From Neural Network Models (Tom Goldstein)

Generating Information From Neural Network Models (Tom Goldstein)

1. Neural nets learn patterns from data. So far so good.

Generating Information From Neural Network Models (Tom Goldstein)

1. Neural nets learn patterns from data. So far so good.
2. But that data is itself not public.

Generating Information From Neural Network Models (Tom Goldstein)

1. Neural nets learn patterns from data. So far so good.
2. But that data is itself not public.
3. Can you observe what a Neural net accepts and rejects and then **infer** the data it was trained on? This is called **an attack**.

Generating Information From Neural Network Models (Tom Goldstein)

1. Neural nets learn patterns from data. So far so good.
2. But that data is itself not public.
3. Can you observe what a Neural net accepts and rejects and then **infer** the data it was trained on? This is called **an attack**.
4. If you own that neural nets, how can you defend against those attacks?

Project Work both sides of the question: find attacks and find defenses against them.

Prereq Some knowledge of Neural Nets.

Learning Fair Representations (Furong Huang)

Learning Fair Representations (Furong Huang)

1. ML is powerful and can help us make decisions quickly!

Learning Fair Representations (Furong Huang)

1. ML is powerful and can help us make decisions quickly!
2. ML can make unbiased choices!

Learning Fair Representations (Furong Huang)

1. ML is powerful and can help us make decisions quickly!
2. ML can make unbiased choices!
3. OH- ML can reinforce prior bias.

Learning Fair Representations (Furong Huang)

1. ML is powerful and can help us make decisions quickly!
2. ML can make unbiased choices!
3. OH- ML can reinforce prior bias.
4. How to make ML algorithms **transparent** so that such bias can be addressed.

Project Make ML's that capture prior data but without the bias.

Prerequisite Math Maturity, Linear Algebra, Prob, Stat, exposure to ML.

Lattice Based Cryptography (Jon Katz)

Lattice Based Cryptography (Jon Katz)

1. Current Cryptography depends on Factoring or other problems in Number Theory to be hard.

Lattice Based Cryptography (Jon Katz)

1. Current Cryptography depends on Factoring or other problems in Number Theory to be hard.
2. Quantum Computers can factor fast; but they don't exist

Lattice Based Cryptography (Jon Katz)

1. Current Cryptography depends on Factoring or other problems in Number Theory to be hard.
2. Quantum Computers can factor fast; but they don't exist yet.

Lattice Based Cryptography (Jon Katz)

1. Current Cryptography depends on Factoring or other problems in Number Theory to be hard.
2. Quantum Computers can factor fast; but they don't exist yet.
3. Clever number theorists and computer scientists are finding ways to factor faster.

Lattice Based Cryptography (Jon Katz)

1. Current Cryptography depends on Factoring or other problems in Number Theory to be hard.
2. Quantum Computers can factor fast; but they don't exist yet.
3. Clever number theorists and computer scientists are finding ways to factor faster.
4. Lattice Based Crypto is a **new** way to do crypto that does not depend on factoring being hard. Algorithms have been developed but need to be tested out, coded up, examined more carefully to see if they are **really** useful.

Lattice Based Cryptography (Jon Katz)

1. Current Cryptography depends on Factoring or other problems in Number Theory to be hard.
2. Quantum Computers can factor fast; but they don't exist yet.
3. Clever number theorists and computer scientists are finding ways to factor faster.
4. Lattice Based Crypto is a **new** way to do crypto that does not depend on factoring being hard. Algorithms have been developed but need to be tested out, coded up, examined more carefully to see if they are **really** useful.
5. Are they useful? You can be on the ones to make it useful!

Project Learn and Code up Lattice Based Crypto Protocols.

Prereq Math Maturity and C/C++ coding ability. Crypto not required.

Activities (Tentative)

1. First week- BEGIN your projects PRONTO!
2. First week- Talks from all the mentors on all the projects.
3. Every Monday- REU lunch. Discussion topics:
 - ▶ Grad School
 - ▶ Bad Science
 - ▶ Work no Math problems together!
4. Every Wednesday- REU talks
 - ▶ From mentors on topics of interest in algorithms.
 - ▶ From mentors on their research
 - ▶ From visitors that happen to be in town.
5. Every weekday- Work on Project.
6. Field Trip to International Spy Museums.
7. There will be presentations of the projects.

Qualifications

1. A course in Algorithms (or a great math background)
2. Knowing how to program
3. See website for prerequisites for some projects
4. US Citizenship for NSF funding. But see next 2 points
5. We have some (not alot!) money for non-citizens.
6. If you are funded for Stipend we may be able to provide Housing.

How to apply: Goto the website!

Will need:

1. Transcript (including Fall 2019)
2. Statement of Purpose
3. Letters from Professors
4. Other stuff that is on form

Logistics

Qualifications

1. **YOU** apply by March 1.
2. **WE accept** or **reject** you. Final decisions made by the End of March (or sooner).
3. If we **reject** you then... Oh well. (Recommend you apply to several REU's and internships.)
4. If we **accept** then you can
 - ▶ **Reject** us! This is perfectly fine.
 - ▶ **Accept** us! And come!
 - ▶ **DO NOT Accept** and then **NOT** come (bad ethics).

If you **ACCEPT** then we look forward to seeing you in June!

Advice for ANY REU-app, Grad School App

1. Apply Early.
2. Get your personal statement done ASAP.
3. Personal Statement should elaborate on what you want to work on, why you are qualified, and why you're interested.
4. Apply to many places (REU programs— Google NSF REU to find more programs).

Quotes from Former Students

From the Summer 2019 REU-CAAR program:

Coming from a small liberal arts college, this program gave me the chance to collaborate with students and professors/postdocs from different backgrounds. The weekly talks given by UMD professors broadened my horizon. REU CAAR helped me be a better researcher and solidified my decision to get a PhD in CS.

Quotes from Former Students

From the Summer 2019 REU-CAAR program:

CAAR was my first REU experience. At first, I faced some difficulties understanding the material but soon enough I caught up on work with the help of my mentor Dr. Gasarch and my colleague Stephanie.

The STEM talks were eye-openers for me in different topics!

The program was also very social. We had three game nights! We had lunches where we worked on fun math problems or had a nice discussion. For me, that helped build relationships faster and helped me know everyone in the program (there were 20 people in the program which I've heard is a lot for an REU program).

The whole experience was amazing, it made me excited for graduate school and I definitely see myself joining another REU in Summer 2020.

More Quotes from Former Students

From the Summer of 2018 REU-CAAR program:

CAAR REU 2018 confirmed my plan to apply to graduate school in computer science and to pursue a research path. I enjoyed everything about this program, primarily the research. I am grateful to be advised by our awesome mentor (John Dickerson) and other phd students. I learned a lot from problem formulation, problem solving, and other aspects of research including collaboration and overcoming challenges.

I also value the time spent with the peers in the REU because they are very inspiring. I am also thankful for the talks that the department organize, which made our summer a very stimulating environment. REU-CAAR is one of the best programs for undergrads who are considering doing research in CS!

More Quotes from Former Students

From the Summer of 2018 REU-CAAR program:

This REU program was really useful for me because it gave me the chance to work as one of the principal investigators on a research problem. When I do research with labs at my university, I normally only get the chance to be involved in the periphery because I'm a student and don't have much experience. In this project, I was involved with all aspects of the project, so I got experience with making decisions and planning the research project. I also enjoyed getting to collaborate with fellow undergraduates, and it was interesting to see a different research environment.

Summary

If you want to

1. Get a research experience
2. Bond with fellow students
3. Get a taste of graduate school
4. Have a great time!

then **APPLY** for REU-CAAR!