David J. Lee

1333 Paresky
Williamstown, MA 01267
✉ *djl5@williams.edu*
djslzx.github.io

Education

2017–2021 **Williams College**, *Williamstown, MA*.
B.A. Computer Science with Highest Honors, Mathematics.
Elected Phi Beta Kappa. GPA: 4.0. GRE: 170vb/170qt.
Thesis: *A Practical Adaptive Quotient Filter*. Advised by Shikha Singh and Samuel McCauley.

Publications

2021 **Telescoping Filter: A Practical Adaptive Filter**, *David Lee*, Samuel McCauley, Shikha Singh, and Max Stein. In Submission.

2021 **Virtual Multicrossings and Petal Diagrams for Virtual Knots and Links**, Colin Adams, Chaim Even-Zohar, Jonah Greenberg, Reuben Kaufman, *David Lee*, Darin Li, Dustin Ping, Theodore Sandstrom, Xiwen Wang. In Submission.

Work Experience

Summer 2021 **Research Assistant, Data Structures**, *Williams College*.
Mentors: Sam McCauley and Shikha Singh.

2020–2021 **Thesis Student, Data Structures**, *Williams College*.
Mentors: Sam McCauley and Shikha Singh.
- Designed and implemented a novel adaptive quotient filter in C/C++.
- Implemented the rank-and-select quotient filter in C, augmented with a theoretically optimal compressed representation for adaptivity.

Summer 2019 **Undergraduate Researcher, Programming Languages**, *Williams College*.
Mentor: Stephen Freund. Part of Synchronicity (NSF Grant #1812951).
- Developed an algorithm using counterexample-guided inductive synthesis to synthesize synchronization disciplines for concurrent programs, leveraging Lipton's theory of reduction.
- Algorithm successfully generated valid synchronization disciplines for medium-sized concurrent test programs.
- Poster: Inferring Synchronization Disciplines to Verify Atomicity of Concurrent Code.

Spring 2019 **Undergraduate Researcher, Knot Theory**, *Williams College*.
- Wrote a combinatorial algorithm in Python to conjecture an upper bound on the number of distinct virtual multi-crossings for a virtual n-crossing, ignoring symmetries.

Spring 2018–Present **Teaching Assistant**, *Williams College*.
Held office/lab hours and graded assignments. 10 hours/week.
- Software Methods (Spring 2020)
- Principles of Programming Languages (Spring 2019, Fall 2019, Fall 2020, Spring 2021)
- Introduction to Computer Science (Fall 2018)
- Data Structures and Advanced Programming (Spring 2018)

Spring 2019 **Tutor**, *Williams College Center for Learning in Action*.
Independently taught introductory computer science to a high school student at Mount Greylock Regional School. 2 hours/week.

Personal Projects

2020 **Learned Bloom Filters**.
- Used PyTorch to develop Bloom filters augmented with neural network models.
- Implemented a single-filter system as described in Kraska's 2018 paper on learned index structures and the "sandwiched" learned Bloom filter from Mitzenmacher's 2018 paper.

2020 **A Peer-to-Peer Privacy-Preserving Location-Based Digital Contact Tracing Protocol**.
- Designed a digital contact tracing protocol that uses GPS data from cellular devices to alert users of potential virus transmission events without compromising user anonymity.
- Simulated in Go using Apache Cassandra.

2019 **Functional Hearthstone in Lisp**.
- Rewrote the Hearthstone game engine in Clojure following functional programming best practices.
- Engine core consists entirely of pure functions that are rigorously tested: mutation is limited to the namespace handling the engine's interface with a web view.

2018 **Augmented Reality Drawing for iOS**.
- Wrote an iOS application that lets users draw curves in 3D space by moving their devices.
- Used ARKit to determine device position from camera data and SceneKit to generate 3D geometries.

Awards

2021 Sam Goldberg Prize. Awarded for the best colloquium in computer science at Williams College.

2021 Computing Research Association (CRA) Outstanding Undergraduate Researcher Award: Honorable Mention.

2021, 2020 Ward Prize (Nominated). Awarded annually to the best student project in the Williams College CS Department.

2020 Phi Beta Kappa (Junior Year). Awarded to top 5% of graduating class by GPA.

Relevant Coursework

Computer Science: Machine Learning, Distributed Systems, Functional Programming, Software Methods, Storage Systems, Theory of Computation, Programming Languages, Algorithms, Computer Organization, Data Structures

Mathematics: Real Analysis, Abstract Algebra, Knot Theory, Cops and Robbers[1], Probability, Linear Algebra, Differential Equations, Multivariable Calculus

Skills

Programming Languages: Python, C/C++, Java, Scala, Rust, Go, Clojure/ClojureScript, Standard ML

Natural Languages: English (Native), Korean (Proficient), Chinese (Intermediate)

[1] A senior seminar on graph theory, with a focus on the game of cops and robbers on graphs.