Christopher Jeong
Portland, OR | 971-517-7475 | chris_jeong@brown.edu | LinkedIn | Portfolio

Education

Ivy League Sep. 2022 – May 2026
Computer Science (BS/MS), Mathematics Science (BA) *City, State*

- **GPA: 4.0**
- Computer Science Coursework: **Data Structures and Algorithms**, **Operating Systems (Graduate)**, **Distributed Systems**, **Computer Networks**, **Computer Systems Security (Graduate)**, **Cryptography**, **Multiprocessor Synchronization**, **Compilers**, Advanced Probabilistic Algorithms (Graduate), Design and Analysis of Algorithms
- Mathematics Coursework: **Statistics**, **Linear Algebra**, **Optimization**, Algebraic Number Theory, Complex Analysis, Galois Theory + Representation Theory, Abstract Algebra, Multivariable Calculus

Experience

Big Bank Jun. 2024 – Aug. 2024
Software Engineer Intern *City, State*

- Developing an internal **React/Java/Spring** project management tool tracking status of product teams in all lines of business as well as project status, version control, and internal mobility opportunities for **1,100 teams** and **35,000** employees. Utilizing **AWS** for scaling and deployment of the application.
- Engineering and maintaining **RESTful APIs** in order to better coordinate API calls between in-house microservices within a **Java** environment. Reworking API calls to coordinate with the team's shift from **Angular** and a private cloud environment to **React** and **AWS**, ensuring consistency in the back-end during the transition of front-end frameworks.
- Creating a **Kafka data pipeline** aggregating data from Jira to derive story progress and making API calls to an internal virtual assistant that automatically notifies employees of story progress. Developing a **Small-Language Model** that allows for real-time analysis of Jira stories.

University Department of Computer Science Jun. 2024 – Aug. 2024
Course Development Assistant *Remote*

- Debugging and stress testing multithreaded **C/x86_64** programs with **10,000+** lines of code, ensuring a seamless fit between file system, virtual memory, threads, and processes, and ensuring that the operating system can handle high-stress situations such as forkbomb attacks and full consumption of disk space.
- Handling the implementation of a **B+Tree** file system to introduce students to more realistic versions of modern file systems.

Game Company Mar. 2023 – Mar. 2024
Software Developer *Remote*

- Initiated a shift from deprecated chatrooms to a new, modernized forum built on **TypeScript**, **Python**, and **SQL**, increasing tournament participation by **30%** and forum visits to **150,000** users yearly.
- Elevated security of user information through **PyCryptoDome**, including password hashing and randomized salting, zero-knowledge proofs to ensure user identity.

University Department of Computer Science May 2023 – Aug. 2023
Undergraduate Research Assistant *City, State*

- Evaluated the intersection of **Natural Language Processing** and **Formal Logic** with application to robotics. Researched Partially Observable Markov Decision Processes and their applications to reinforcement learning
- Restructured existing **React/CSS/Node/MongoDB** web demonstration of the project by integrating leaflet.js, allowing users to give instructions to a robot and demonstrating how the instructions translated to linear temporal logic.

Projects

Lenstra's Elliptic Curve Factorization Algorithm | *Python, Cryptography, Algorithms* Apr 2024 – May 2024

- Meshed principles of multiprocessor synchronization, algebraic number theory, and elliptic curves to implement a factorization algorithm of **RSA** moduli in sub-exponential time. Factors **128-bit** moduli in 61 seconds.

Multiple Candidate Voting Protocol | *C++, CryptoPP, Cryptography, SQL* Apr 2024 – May 2024

- Formulated a heavily mathematical protocol in **C++** that allows voters to anonymously vote for candidates using zero-knowledge proofs, ensuring votes and user information remain secure during network communication.
- Utilized **SQL** to store and query encrypted user information and implemented custom serialize/deserialize functions for user information.

DIY OS | *C, x86_64, Unix, Python* Jan 2024 – Apr 2024

- Engineered a Unix-based operating system kernel from scratch that can run **C** programs. Implemented processes, threads, mutexes, virtual memory, physical memory, page tables, system calls and disk management.

Technical Skills

Languages: Java, Python, C/C++, SQL, JavaScript, TypeScript, HTML/CSS, OCaml
Technologies: React.js, Node.js, Flask, Kafka, JUnit, MongoDB, Spring, Jenkins, Maven, AWS, SQL, LiquiBase
Developer Tools: Git, Docker, VS Code, Postman, GitHub, BitBucket