

Qingyi (Luna) Lu

Box 1910, Brown University, Providence, RI 02912-1910
Email: qingyi_lu@brown.edu | Cell: 315-461-7977 | Website: qingyilu.com | LinkedIn: linkedin.com/in/qingyilu97

EDUCATION

Brown University - Providence, RI | Overall GPA: 4.0/4.0

Expected May 2021

- Master of Science in Computer Science
- **Relevant Coursework:** Distributed Systems at Scale: Microservices Management, Operating Systems (C), Distributed Computer Systems (Golang), Special Topics in Computational Linguistics, Deep Learning (Python), Computer Graphics (C++, OPENGL)

Hobart and William Smith (HWS) Colleges - Geneva, NY | Overall GPA: 3.97/4.0

May 2019

- Bachelor of Science degrees with Highest Honors in Mathematics and Computer Science
- **Honors Thesis:** "Machine Learning for Phylogenomics: Improving Statistical Binning Technique for Species Tree Reconstruction"
- **Honors:** Summa Cum Laude, Member of Phi Beta Kappa, Catherine Adele Rippey '35 Prize in mathematics and Computer Science, Robert Beinert Prize in mathematics, William Ross Proctor Prize in Mathematics
- **Relevant Coursework:** Artificial Intelligence(Java), Database Theory and Practice(SQL), Software Development(Java)

TECHNICAL SKILLS

- **Language:** Java, C/C++, Python, Go, JavaScript, SQL, R, HTML, Ruby, CSS
- **Tools/Frameworks:** Docker, Git, React, Node.js, Express, Anaconda/Python3, Qt, Arduino, TensorFlow, PyTorch, MATLAB
- **Database:** MySQL, PostgreSQL, MongoDB, Firebase

PROFESSIONAL EXPERIENCE

Teaching Assistant | Brown University, Department of Computer Science, Providence, RI

Aug. 2020 - Present

- Designed and reconstructed the format of course assignments and projects in Python
- Added new unit tests into auto-grader, and prepare lecture slides, course website and other supplemental materials
- Graded assignments, held weekly office hours, lab hours and piazza hours to debug and answer questions

Software Engineering Intern | Global Market Division, Goldman Sachs, New York, NY

Jul. 2020 - Aug. 2020

- Developed and maintained enterprise level trading platform named Equities Custom Basket used by the trading team, which combining any specific sets of equities products and trade it as one product
- Built the front-end UI of the platform with React.js
- Implemented sever-side backend microservice using Java Spring Boot Framework, Spring MVC model along with PostgreSQL as database for RESTful API
- Enhanced the system infrastructure with large scale data written in Slang/SecDB by enabling a tool on trading platform that allows to rebalance the pairs of Custom Basket with cash component, and performed the analysis to reduce the risk
- Set up a trading tool in the system infrastructure to view hedge creations associated with pairs of Custom Basket
- Built a library to create the upcoming corporate actions report for internal and external baskets and sent to clients as an email

Research Assistant | Mathematical Phylogenetic Scholars Program, HWS Colleges, Geneva, NY

Aug. 2016 - Aug. 2019

- Built own R package for Booster, the alternative bootstrap estimation method, and used in the fundamental step in binning pipeline to produce more accurate species trees
- Applied the statistical theories to design the algorithms in order to obtain precisely classification
- Implemented theoretical binning pipeline in R and Python, and performed data analysis for the large sets of sequence data
- Developed console management application for data storage and synchronization in a large dispatch center
- **Publication:** Joseph P Rusinko, Jennifer Vandenbussche and Qingyi Lu, "Improving Statistical Binning Techniques for Species Tree Reconstruction"

PROJECTS

PuddleStore Distributed Filesystem | Class Project, Providence, RI

Apr. 2020 - May 2020

- An application written in Golang that implements the distributed filesystem based on the Tapestry and Zookeeper
- Designed and implemented a file system to store and replicate all data in memory by using Tapestry and put all file inodes in Zookeeper, which allows the users to do the files and directories operations
- Implemented the copy-on-write update strategy to maintain data consistency and a synchronous update system to update pointers to data, and applied the Zookeeper distributed file locking to avoid herding problem and incorporate the fault tolerance

Weenix Operating System | Class Project, Providence, RI

Feb. 2020 - May 2020

- A project written in C that implements the Weenix OS, parts of a Unix kernel
- Implemented threading libraries for both N-1 and M-N models to support user-level threads running in parallel on multiprocessor
- Built VFS and S5FS that provides an interface between the OS kernel and various file systems which allows one to add many different types of file systems to one's kernel and access them through the same UNIX-style interface
- Constructed additional virtual memory to make the kernel manages user address spaces, runs user-level code, and services system calls

RNN-based Classical Chinese Poetry Generation | Class Project, Providence, RI

Nov. 2019 - Dec 2019

- A Python application that reimplements and improves the existing RNN-based encoder-decoder mechanism to auto-generate fluent and thematic-consistent classical Chinese poems based on the input topics by applying the *Chinese-poetry* dataset
- Constructed the encoder-decoder based model with poem planning strategy by building three separate model blocks: Word Poem Block, Sentence Poem Block, and Context Poem Block, to firstly generate topics from input, then produce the first sentence from topics and finally generate other three sentences each sentence
- Improved the poem generation pipeline by using Transformer and evaluated the model by using the BLEU-2 Score Evaluation