Nathan Chung

Phone: (240)-750-8621, Email: nathanchung80@gmail.com Github: github.com/nathannchung

EDUCATION

University of Maryland College Park

College Park, MD

Bachelor of Science, Computer Science (Data Science)

August 2019 — May 2023

- Honors: Science, Discovery, and the Universe Scholars Program (3% acceptance rate for applicants)
- Organizations: Alpha Kappa Psi Professional Business Fraternity
- Selected Coursework: Object-Oriented Programming, Computer Systems, Programming Paradigms, Algorithms, Discrete Mathematics

EXPERIENCE

JP Morgan Chase Remote

Software Engineering Virtual Experience

July 2020 — *August* 2020

- Designed a dashboard to allow equity traders to visualize and monitor trading strategies for live stock price feeds.
- Leveraged JP Morgan's open source visualization framework, Perspective, to support stock filtering and aggregation functionality.
- Implemented calculation pipelines to compute metrics such as bid-ask spreads and simple moving averages in real-time.

PROJECTS

WordNet Semantic Graph | Ruby

- Constructed a Directed Acyclic Graph (DAG) to represent the semantic relationship between words.
- Architected semantic graph to group words into synonym groups (synsets), and map "is-a" relationships from hyponyms (child synset) to hypernyms (parent synset). For example,
- Developed a command parser to allow users to perform actions on the semantic graph such as word lookup, least common ancestor search, and hyponym-hypernym relationship load.

Unix Shell Reconstruction | C, Unix, Emacs

- Re-Engineered a Unix shell to support input/output redirection, command piping, and command chaining via the "&&" operation.
- Created a parser to parse and validate user input, and an executor to execute commands and apply subshell forking when necessary.

SKILLS

Technical Skills: Java, C, Python, Ruby, OCaml, Assembly (AVR)

Tools: Git, Emacs, Unix, Windows, Excel, Powerpoint, Word

Languages: Native English, Limited Working Proficiency in Korean

Interests: Music Production, Fashion, Soccer