Willis Wang

Software Developer

wwang4292@berkeley.edu **williswang.dev** **github.com/willislwang** **(408) 966-2909**

Education

B.A. Computer Science **UC Berkeley**

3.74 GPA August 2018 - May 2022

Courses taken: Operating Systems, Algorithms, Networking, Databases, Data Structures, Machine Structures, Artificial Intelligence, Discrete Math, Probability Theory

Experience

Undergraduate Researcher **Video and Imaging Processing Lab**

August 2020 - Present Berkeley, CA

- Collaborated to develop a multi-modal indoor proximity detection framework with Professor Zakhor
- Design and implement magnetometer trace batching, feature extraction, and distance classifier
- Designed for use in COVID-19 contact tracing

Software Engineer Intern **8th Wall**

May 2020 - August 2020

- Worked with AWS to improve existing infrastructure to allow for safe, simple development and testing
- Streamline development process by building automated database migrations and CDN updates synchronized to git
- Design and build customer repository backup and restoration framework

CS182/282A Reader **UC Berkeley**

Janurary 2020 - May 2020 Berkeley, CA

- Reader for upper division Neural Network course CS182/282A
- Assist in assignment release and rubric creation, grading both homeworks and exams

CS61B Academic Intern **UC Berkeley**

September 2018 - December 2018 Berkeley, CA

- Help over 40 students gain a deeper understanding about core data structures, abstract data types, algorithms, basic principles of software engineering, and the Java programming language
- Strong foundation in OOP, Dynamic Programming, Graph Traversal, Data Structures, and Sorting Algorithms
- Hosts weekly reviews focused on conceptual applications

Projects

Delta Lake Caching

October 2020 - Present Scala

- Research project for graduate level class CS 262A: Advanced Topics in Computer Systems
- Work with Databricks engineers to design and implement a read/write caching layer for Delta Lake using RocksDB

PintOS

Feburary 2020 - May 2020 C

- A simple operating system framework for the 80 x 86 architecture
- Implemented kernel syscalls, processes, scheduling, filesystems, caches, etc.

TSP Variation Approximator

December 2019 Python

- Write an optimal solution finder for a version of the TSP problem
- Uses a combination of approximation algorithms such as Chistofides, k-means clustering, and local search to find a valid and cost-efficient solution

Languages

Programming Languages: **C/C++, Java, Python, SQL, HTML/CSS, Javascript (Node, Express)**

Operating Systems: **Linux, macOS, Windows**

Languages: **English**