# Zuwen (Dave) Gan

415-691-0832 • zgan@dons.usfca.edu 2845 Golden Gate Ave Apt 5 San Francisco, CA 94118 GitHub Page: zgan.github.io

## **OBJECTIVE**

Seeking an internship as a software engineer.

## **TECHNICAL SKILLS**

- Programming Skills: Java (proficient), C, HTML/CSS, Python
- Environments: Eclipse, Adobe Dreamweaver
- **Revision Control**: Subversion, GitHub

## **EDUCATION**

University of San Francisco, San Francisco, CA

- Bachelor of Science Degree, Computer Science
- Major GPA: 3.68, General GPA: 3.62

### **EXPERIENCE**

 Software Development Course Teaching Assistant, USF
 Spring 2015 - Present

 • Hold office hours to help students with questions, assist with grading, and developing solutions and test programs. Proactively, learn new technologies and programs such as JSON, can answer student questions.
 Spring 2015 - Present

 Introduction to Computer Science II Course Teaching Assistant, USF
 Spring 2015 - Present

 • Hold office hours to answer student questions and assist with grading.
 Spring 2015 - Present

 • Hold office hours to answer student questions and assist with grading.
 Spring 2015 - Present

 • On average, assist up to 10 students a day from all CS courses.
 Spring 2015 - Present

August 2012 – May 2016 (Expected)

#### **PROJECTS**

#### Search Engine (Java, Software Development)

Build a thread-safe inverted index with multithreaded search function to store data. Build a website as interface for the search using MySQL database language to store user information, such as visited results and search history; work with cookie and session to track user status and write servlet codes with Java.

#### MPI-Finding Primes (C, MPI Library)

Use MPI to parallel the process of finding primes with the given range. Each process finds it's own primes and send to the reciver, which is right shifted by the amount of the bitmask. And the recivers use merge-split to get a merged list until process 0 receives all the primes from other processes.

#### Finding Shortest Path (Java)

The project is about using Dijkstra algorithm to calculate the shortest path between two cities and priority queue to increase the efficiency. There is a GUI for showing the algorithm, which is a map with cities on it. User can click any two cities and then it will calculate the shortest path between two cities.