Site(In Progress)

bren-a.github.io

Contact Information

Brendan Alger
balger97@ucla.edu
(818) 915-3695

Address

425 University Ave
Burbank, CA 91504
United States

Relevant Courses Taken

- Data Structures
- Computer Organization(Assembly)
- Software Construction Lab
- Algorithms
- Applied Phonetics
- Syntax
- Phonology
- Semantics

Programming Languages

- C/C++(Proficient)
- HTML/CSS(Beginner)
- Python(Familiar)
- Java(Prior Exp.)

Certificates

- CPR/AED
- First Aid/ Title 22
- OHSA

Skills

- Bash
- LaTeX
- MS Office
- Leadership
- Communication

Languages

- English
- Tagalog
- French

Brendan Alger

Student

Education

Graduation Jun, 2019: University of California, Los Angeles

Bachelors in Linguistics and Computer Science

Experience

Jun 2017 - present: UCLA Recreation

Lifeguard, Jun 2017 - Apr 2018; *Head Lifeguard*, Apr 2018 - present

- Assists in development and leading of team training exercises to foster teamwork and review necessary skillsets
- Versed in the facility rules and regulations of 6 different pools to properly inform and enforce patrons and ensure the safety of everyone
- Mediate any disputes and issues among patrons
- Observe up to over 100 people in a lively environment and proactively identify and take measures to prevent hazards and ensure safety

Projects

2018, *OpenMP Optimization*

Optimized an image processing deblurring alogorithm

- Used strength reduction, loop unrolling, loop tiling, common subexpressions, and OpenMP to decrease runtime.
- Resulted in an average speedup of 8x compared to the original program.
- Optimized the program for a class assignment and received a score of 100.

2018, *Ceasar Cipher*

Designed a program in C++ that can crack a Ceasar Cipher
Project Details:

- Finds isomorphs using a wordlist and maps it on a hash map whose previous states are saved in a stack via recursion.
- Created a hash map table that uses a dynamically allocated array of node pointers to a binary search tree (Did not use the C++ STL hash map).
  - Shortened insertion and search to O(1) in most cases using a hash map.
  - If N items shared the same key, searching was on average O(logN) thanks to using a binary search tree instead of a linked list.

2018, *NachenBlaster*

Developed an object-oriented 2D space-shooter game in C++ using FREEGLUT

- Used polymorphism to cut 100s of lines of code and simplify debugging