

Christine Le

(714)-467-8969 | christinele1301@gmail.com | [linkedin.com/in/christine-le-49458423b](https://www.linkedin.com/in/christine-le-49458423b)

EDUCATION

University of California, San Diego

B.S. in Computer Science and Mathematics

La Jolla, CA

Sept. 2023 – June 2027

Relevant Coursework: Java Programming I, Basic Data Structures & OO Design, System Programming and Software Tools, Discrete Math & Graph Theory, Advanced Data Structures

TECHNICAL SKILLS

Languages: C/C++, Java, Python, JavaScript, HTML/CSS, Bash

Frameworks: React, Node.js, Next.js, JUnit, Express

Developer Tools: VS Code, Vim, Eclipse, Git, GitHub, Postman, GDB, Valgrind, Make, Linux/Unix

Libraries/Databases: pandas, PIL, Matplotlib, MongoDB

PROJECTS

Graph Algorithms | *C++, STL, Makefile*

July 2025

- Implemented an undirected Graph class in C++ with adjacency-based data structures supporting graph construction and traversal.
- Developed algorithms for BFS, Dijkstra's shortest path, connected components, and clustering thresholds with optimized performance.
- Ensured correctness and maintainability through unit tests, modular design, and automated builds using Makefiles in a Linux environment.
- Applied algorithmic problem-solving and data structure design principles to strengthen understanding of graph theory in practice.

Huffman File Compression Tool | *C++, STL, Makefile*

July 2025

- Implemented Huffman compression algorithm in C++ to reduce file storage requirements, achieving lossless compression for files up to 10 MB in size.
- Developed complete encode/decode pipeline with custom binary tree construction, frequency analysis, and optimal bit-code generation for character mapping.
- Built robust file I/O system with header-based metadata storage enabling perfect reconstruction of original files with zero data loss.
- Created paired compression/decompression programs with command-line interface supporting arbitrary file formats and automated build system using Makefiles.

Malloc Simulator | *C, Linux, gdb, Valgrind*

February 2025

- Developed custom memory allocation system implementing malloc() and free() functions with best-fit allocation policy and 16-byte alignment.
- Built dynamic memory management with block headers/footers, coalescing algorithms, and block splitting for optimal heap utilization.
- Implemented pointer arithmetic and bitwise operations to traverse heap blocks, manage metadata, and handle memory fragmentation.
- Created comprehensive testing suite with edge cases, stress tests, and memory validation using assert statements and custom heap visualization.

Pioneer Shell (Pish) | *C, Linux, gdb, Valgrind*

February 2025

- Developed a Unix-style command-line shell supporting interactive and script execution modes with command parsing and process management.
- Implemented built-in commands (cd, exit, history) with error handling and persistent command history storage in ~/.pish.history file.
- Built process management system using fork(), execvp(), and wait() system calls to spawn child processes for external program execution.
- Created robust input parsing with whitespace handling using string manipulation functions to break commands into argv arrays.