THOMAS CHAN
15 Caulfield Rd, Wayland, MA 01778 | (617) 838-9803 | thomas.chan@tufts.edu

---

COURSEWORK

COMP15: Data Structures
Language: C++
Homework 1
- ➔ Implemented string manipulation using sequences
- ➔ Functions included removing at a specific index, insertion alphabetically, and concatenation of strings

Homework 2
- ➔ Implemented string manipulation using linked lists
- ➔ Functions included removing at a specific index, insertion alphabetically, and concatenation of strings

Homework 3
- ➔ Implemented a reverse polish notation calculator using a stack

Homework 4
- ➔ Implemented an alphabetizer using a binary search tree

Homework 5
- ➔ Implemented insertion, merge, and quick sort
- ➔ Analyzed time complexity for each algorithm

Project 1
- ➔ Implemented a warehouse simulation, handling packages for delivery
- ➔ Functions included simulating the use of conveyer belts equally versus using conveyer belts as they are available for use

Project 2
- ➔ Implemented a search function for a given file tree using a hash table
- ➔ Functions included a case-sensitive and case-insensitive search

COMP40: Machine Structure and Assembly Language
Language: C
Homework 1
- ➔ Implemented a program that calculates the average brightness of a grayscale image
- ➔ Implemented a program that prints "similar lines" from a file tree in groups

Homework 2
- ➔ Implemented a program that checks if a sudoku puzzle solution passed as a graymap file is valid
- ➔ Implemented a program that removes black edges from photocopy passed as bitmap files

Homework 3
- ➔ Implemented an image manipulator using portable pixmap files
- ➔ Functions include rotating, flipping, and transposing by row-, column-, or block-major
- ➔ Effects of locality/cache performance was for functions called with row-, column-, and block-major

Homework 4
- ➔ Implemented an image compressor and decompressor for portable pixmap files

- → Using cosine transformations between RBG and video component and bit packing to handle compression

Homework 5
- → Analyzed assembly code to defuse Bryant and O'Halloran's binary bomb and secret phase
- → Wrote code for personal interpretation for each phase of the bomb

Homework 6
- → Implemented a Turing complete, 32-bit Universal Machine
- → Handled 14 instructions, for register and segmented memory manipulation

Homework 7
- → Optimized previously implemented Universal Machine
- → Analyzed resulting assembly code to see if further improvement possible

Homework 8
- → Implemented reverse polish notation calculator in HTML

You may contact me at ThomasLChan@gmail.com if you would like to view the source code or inquire about lab work. Given Tufts' academic integrity policy, the repositories holding my solutions are private.