Holesum Design Plan

170D WOBC: Module L Exam II (A)

CW2 Kyle Spicer

Due Date: 22 November 2022

1 Project Summary

1.1 Description

In this exercise you will write a program to identify ”holes” in a data file and report the number of holes and the size of the largest hole.

A data file is an ascii text file containing an m x n matrix of zeros and ones, where holes are groups of zeros such that any zero in a group is adjacent to at least one other zero in the group. A single zero is implicitly adjacent to itself so a hole can be of size one.

Write an executable application that reads in a single ascii text file and reports the number of holes and the size of the largest hole.

2 Architecture

2.1 Directories

2.1.1 holesum (top level directory)

- Makefile
- holesum_driver.c (provided by instructors)

2.1.2 src

- holesum.c (main program to complete logic for program)
- holesum_functions.c
- holesum_functions.h

2.1.3 test

- all test .txt files (provided by instructor)

2.1.4 doc

- design.pdf
- writup.pdf
- testplan.pdf
- holesum.1

2.2 Functions

- void improper_argc_print_and_exit(): displays error and program usage message
- void array_print(int **array, int rows, int columns): visually displays 2D array
- void array_destroy(int **array, int rows): frees allocated memory for 2D array
- int dfs(int **array, int outer, int inner, int row, int column): conducts dfs in 2D array
- void display_all_hole_sizes(char *argv[]): runs program and prints size of all holes

3 Program Flow

1. holesum.c begins by receiving command line arguments from the user.
2. after the command line arguments are verified, the file is opened
3. reads the entire file to capture node amount, number of rows and columns.
4. The file pointer is reset to the beginning of the file.
5. memory is allocated for 2D matrix using nested for loop using rows and columns size
6. nested for loops begin iterating through the matrix looking for 0's
7. If a 1 is encountered, the program continues to the next position
8. If a 0 is encountered, a depth first search is conducted to find connecting 0's
9. The 0 is turned into a 1 (marking it visited) and size of hole is returned
10. The hole size is compared with previous holes and largest hole is recorded
11. Additionally, there is a counter to track amount of holes in each matrix
12. Once complete, holesum.c provides two print statements for the amount of holes found and the size of the largest hole.
13. After the holesum executable is created, compile and run with the holesum_driver.c.
14. If a valid ascii file is sent trough the program, the driver code will verify the output of the holesum executable and display test results.

4 Extra Credit Items

- create manpage holesum.1
- program outputs size of all holes found in current search. (logic is commented out because program wouldn't pass with it displaying).