Willia Hardy
November 28, 2018
Data Structures Project
Student ID: 2905326

Project 4 Report

Summary of Project

This project was intended to use the Union-Find Problem and a Breadth-First search to smash the walls of a maze until a path is created. This program should allow user input to determine the dimensions of the maze. The program will then pick random numbers to smash the walls of the maze until a path is formed. Logically, each time a wall is smashed, based on a neighboring condition, those cells are unionized, and they now belong to the same set.

Manifest

- DisjSets.h
  - Code given from website for the DisjSets class
- maze.h
  - Code for the functionality of the maze
- mazeCell.h
  - Code given from website that has inline methods for mazeCell class
- DisjSets.cpp
  - Code to implement the methods in DisjSets.h, especially the find method
- maze.cpp
  - Code to implement the methods in maze.h
- buildmaze.cpp
  - Main file that prints, smashes the maze walls, and displays the iteration of the maze
- CMakeLists.txt/./buildmaze
  - File used to compile and run the project
  - I used the “cmake ..” command and “make” in order to run the program
  - Or the g++ command to compile the project using current c++11 features. (**g++ -std=c++11 buildmaze.cpp DisjSets.cpp maze.cpp -o buildmaze**) then use **./buildmaze** to run the program

Answers to Questions

1. What does it mean for two cells to “be connected” with respect to this maze?
   a. For two cells to “be connected” means that they belong to the same set and they have the same root node. The cells become apart of the same set after the union is performed on the neighboring cells.