Justin Gnatiuk
Data Structures
Project 4 Report

Summary

For our fourth data structures project, we were to build a maze of inputted size that generated a path from the starting cell (top left) to the bottom cell (bottom right) of the maze.
This project utilized a disjointed set data structure to randomly choose two cells and determine if they are neighbors as well as check if the two cells are connected.
The maze functions are what proved most difficult to me as I have never worked with one before.

File Manifest

1. maze.h : Header file containing maze object
2. mazeCell.h : Header file containing mazeCell object
3. disjSets.h: Header file containing disjointed sets data structure
4. disjSets.cpp : File containing function definitions for the disjointed sets data structure
5. maze.cpp : File containing function definitions for the maze object
6. BuildMaze.cpp : File containing driver function for the project.
7. Project4.PDF: Project report

Questions

What does it mean for two cells to "be connected" with respect to the maze?

> for two cells to "be connected" means that they belong to the same set. With regard to the maze, two separate cells may not be neighbors, in the sense that they are not directly touching each other on the grid, but they can still be connected, meaning they could still be part of the same set.
As the maze randomly chooses cells it is destroying the wall between them if they are neighbors (if they are directly touching each other) and at the same time updating the disjointed set and merging those two cells as part of the same set.
Further down the line, two cells may not be neighbors, but they could still be apart of same set if a path of walls has been knocked down between them, the maze itself is just a visual representation of the disjointed set updating and merging sets until the top left cell and bottom right cell are in the same set.