Anshul & James
Project 2 Report
Date: 7/23/2021
CECS 451

- This program solves a 2D maze with the search algorithms BFS and DFS. It takes an input file containing information about the maze and we need to perform each search algorithm, so we can compare and see the differences between both algorithm.

- Breadth-First Search (BFS) is an algorithm that can be used to find a path between randomly-chosen start and goal nodes. BFS is implemented with a queue. It starts by visiting a node, then adding all of that node's neighbors into the queue. Each time it adds a neighbor to the queue, it adds a pointer that references the neighboring node's parent. The reason for this is that it doesn't actually construct the path during its initial traversal of the graph. It repeats the process of visiting the nodes in the order they entered the queue until the queue is empty, which means all nodes have been visited, or until the goal node is found, since I use an early exit in my implementation. Once the traversal is complete, it starts at the goal node and follows the pointers until it reaches the start node.

- Depth-First Search (DFS) is one alternative to BFS. Instead of visiting each neighbor before considering their children, it will plunge all the way to the bottom of the maze, continually choosing the first child, before retracing its steps, visiting children on the way up. This behavior is clearly seen in the output, as it always chooses to go left until there are no children left. Output illustrates this nature as there are long distinct paths generated rather than the broad search performed by BFS.

```
Breadth First Search:
%%%%%%%%%%%%%%%%%%%%%%%%
% %%        % %      %
%    %%%%%% % %%%%%% %
%%%%%%      ...%      %
%    % %%%%%%.%% %%%%%
% %%%% %     ..   %   %
%        %%%.%%%   % %
%%%%%%%%%%... %%%%%% %
%..........%%        %
%%%%%%%%%%%%%%%%%%%%%%%%
Path taken:  [77, 78, 79, 101, 123, 122, 144, 166, 165, 164, 186, 185, 184, 183, 182, 181, 180, 179, 178, 177]
Path cost:  20
Number of vertices expanded:  94
Maximum size of the fringe:  9

Depth First Search:
%%%%%%%%%%%%%%%%%%%%%%%%
% %%        % %      %
%    %%%%%% % %%%%%% %
%%%%%%      ...%      %
%    % %%%%%%.%% %%%%%
% %%%% %     ....%...%
%        %%% %%%...%.%
%%%%%%%%%%....%%%%%%.%
%..........%%........%
%%%%%%%%%%%%%%%%%%%%%%%%
Path taken:  [77, 78, 79, 101, 123, 124, 125, 126, 148, 149, 150, 128, 129, 130, 152, 174, 196, 195, 194, 193, 192, 191, 190, 189,
167, 166, 165, 164, 186, 185, 184, 183, 182, 181, 180, 179, 178, 177]
Path cost:  38
Number of vertices expanded:  37
Maximum size of the fringe:  8
```

- In this maze-setting DFS is faster than the BFS algorithms measured in wall clock time and the Number of vertices expanded in almost less than half for the DFS as compared to BFS but DFS does more work. However, that path is usually not the shortest. We can say that, BFS always finds the optimal path.