# CS 211 PA #2

In this assignment you will write a program that solves a maze by finding a connection between two locations within the maze.

## Maze Text File

Our maze will be read in via text file. The first line informs us of the width and height of the maze. Each line afterward describes an entire line in the maze. Each "space" in the maze is represented with one of the following symbols:

| | |
|---|---|
| # (pound) | Wall |
| . (period) | Open space |
| o (lower case letter o) | Starting location |
| * (asterisk) | Ending location |

Below is an example of a 7x4 maze.

```
7 4
#######
#...#o#
#*....#
#######
```

## Program Specification

Your program will read in any arbitrary maze file and find a solution (if possible) using both breadth-first and depth-first search. If the maze is solvable, your program will generate two output files: solution_dfs.txt (solution when using DFS) and solution_bfs.txt (solution when using BFS). You will use the lower-case character "x" to denote where your maze solver has walked. Below, I provide possible solutions for the maze example above:

DFS Solution

```
#######
#xxx#o#
#*.xxx#
#######
```

BFS Solution

```
#######
#..x#o#
#*xxxx#
#######
```

Note that depending on how you implement your searches, **you may get slightly different answers**. This is okay as long as what you output are valid depth-first and breadth-first searches. If the maze is not solvable (i.e. there is no path between start and finish), simply output "no solution to the screen."

## Header Comment, and Formatting

1. Be sure to modify the file header comment at the top of your program to indicate your name, student ID, completion time, and the names of any individuals that you collaborated with on the assignment.
2. Remember to follow the basic coding style guide. A basic list of rules is included with this document.

## Reflection Essay

In addition to the programming tasks listed above, your submission must include an essay that reflects on your experiences with this homework.  This essay must be at least 350 words long.  Note that the focus of this paper should be on your reflection, ***not*** on structure (e.g. introductory paragraph, conclusion, etc.).  The essay is graded on content (i.e. it shows deep though) rather than syntax (e.g. spelling) and structure.  Below are some prompts that can be used to get you thinking.  Feel free to use these or to make up your own.

- Describe a particular struggle that you overcame when working on this programming assignment.
- Conversely, describe an issue with your assignment that you were unable to resolve.
- Provide advice to a future student on how he or she might succeed on this assignment.
- Describe the most fun aspect of the assignment.
- Describe the most challenging aspect of the assignment.
- Describe the most difficult aspect of the assignment to understand.
- Provide any suggestions for improving the assignment in the future.

## Deliverables

You must upload your assignment through Canvas no later than midnight on Saturday, February 17, 2018.

## PA #2 Checkin

During lab on 2/12 you must demonstrate your program to your instructor.  To receive full credit, your program must be able to parse a maze into a 2D vector.

## Grading Criteria

Your assignment will be judged by the following criteria:

### Reflection essay (5pts)
- Your reflection meets the minimum requirements as specified earlier in this document.

### PA Checkin (10pts; due 9/25)
- Your program correctly parses a supplied maze file into object form ready for searching.  Note that you don't have to have the search function implemented.

### Style (10pts)
- Your project contains good structure and implements the required classes.  Your program intelligently uses classes when appropriate and generally conforms to good OOP design (i.e. everything isn't slapped into main).

### Output Files (15pts)
- Your program correctly generates BFS and DFS result files.

### Path Finding(30pts)
- Your program correctly finds a valid path from a starting point to the specified end point using both depth-first and breadth-first search.
- Your program correctly identifies unsolvable mazes.

## Grade Distribution

Your final grade for the assignment will be determined based on the number of points earned.  Note that failing to use good design (e.g. objects, appropriate data structures), regardless of score earned, may result in a lower overall grade.

| Final Percentage | Percentage Required |
|------------------|---------------------|
| 100              | 90                  |
| 90               | 85                  |
| 80               | 75                  |
| 70               | 60                  |
| 60               | 45                  |
| 50               | 30                  |
| 40               | 25                  |
| 25               | 20                  |