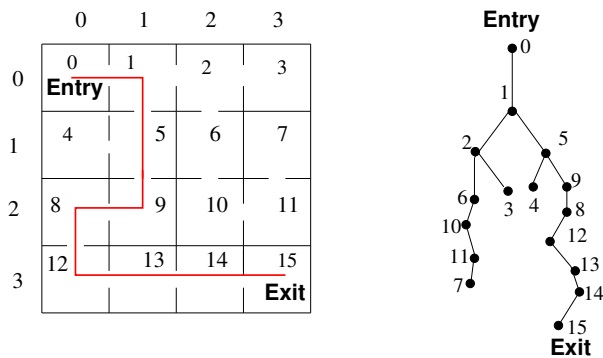


# CPSC 221 Final Project

due Monday, August 6

- Problem description.

Find a path from the entrance to the exit in a maze that consists of rooms connected by doors. Each room is enumerated by consecutive integers, from 0 to  $n^2 - 1$ , and it can have no more than 4 doors oriented to *North*, *South*, *East* or *West*. The entry to the maze starts at the room 0 and ends at the room  $n^2 - 1$ . The maze can be modeled by an undirected graph, where a vertex corresponds to a room and an edge corresponds to a door, see the maze below with the path and the corresponding graph for the maze.



- Use an adjacency linked list or adjacency matrix to represent the graph for this problem.
- Design an algorithm to find a path from the entrance to the the exit of the maze.
- Display the graph in the text format, use zero to denote a vertex (room) on the entry-exit path and x otherwise. Provide the length of the path.

```

o  o  x  x
x  o  x  x
o  o  x  x
o  o  o  o

```

The length of the path above is 8 which is the number of the graph edges (doors to open) to reach the Exit.

- Programming instructions:
  - Demonstrate a good C++ programming style; for more details, see the link.
  - Load the graph data from an input file. Your program should be able to read graph data from any user file (e.g. ask about the file name, or get it from a command line argument). The format of the graph data file is given below (the comments here are for explanation only and should not be in the actual data file):

```

16  /* THE NUMBER OF ROOMS */
/* DOORS LISTED IN ORDER: NORTH, EAST, SOUTH, WEST FOR EACH ROOM: 0 = NO DOORS */
0 1 0 0
0 1 1 1
0 1 1 1
0 0 0 1
0 1 0 0
1 0 1 1
1 0 1 0
0 0 1 0
0 1 1 0
1 0 0 1
1 1 0 0
1 0 0 1
1 1 0 0
0 1 0 1
0 1 0 1
0 0 0 1

```

- Use an adjacency list to organize the input data as a graph.
- Implement an algorithm that finds the entry-exit path.
- Implement a function that displays the path.
- Create a menu that includes the following options:
  - Print the adjacency list for a given graph.
  - Find the length of the entry-exit path generated by your algorithm.
  - Print out all vertices (room numbers) on the entry-exit path.
- Provide at least two test cases, each consisting of a maze data, an input file and an output file in text format, to verify the correctness of your program.
- (Bonus: 10 points) Find all possible entry-exit paths and display them in text format (as above). Decide which one is the shortest path by providing the length for each path.
- Write a report that includes
  - data structures used to represent the graph and the graph operations.
  - your algorithm along with the theoretical analysis of its running time in terms of Big-O asymptotic notation.
  - the test cases for verifying the correctness of your program.
- Submit all source files and a PDF version of the report to eCampus by August 6.
- Submit a hard copy of the cover page, report and the source code. Put your name, section number and user name in the header of the report and the source file in the labs to your TA on August 7.