Algorithms and Analysis – COSC2123/1285

Assignment 2

Dr. Yongli Ren

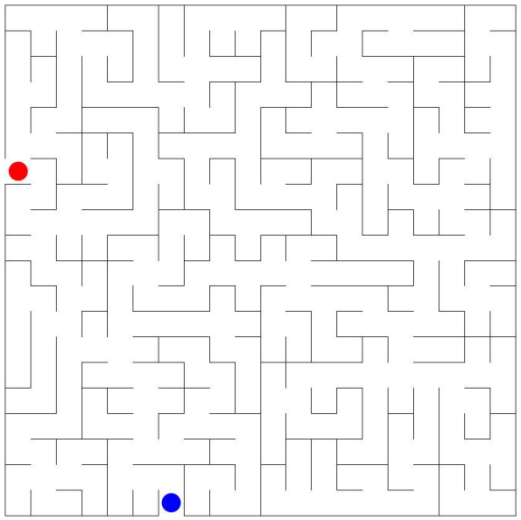
(yongli.ren@rmit.edu.au)

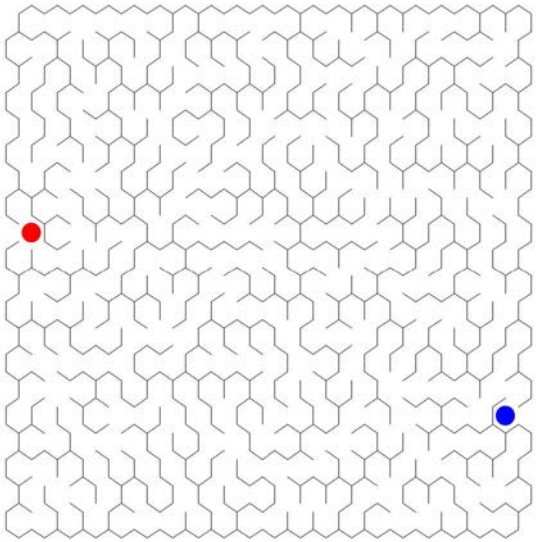
Computer Science & IT

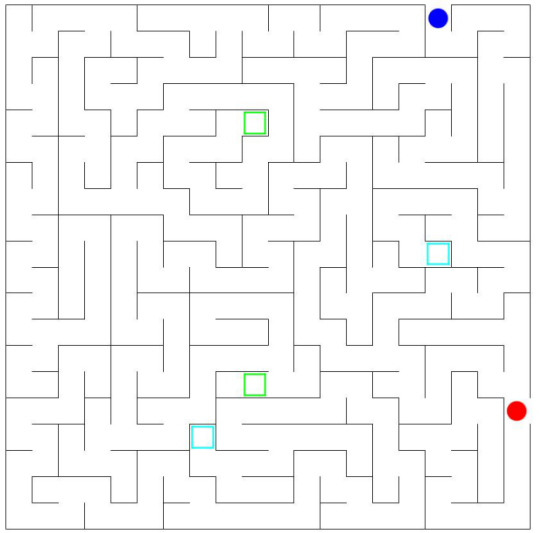
School of Science

Assignment 1

- Due date:
  - 11:59pm Sunday, 15th Oct 2017
- Weight: 15%
- Pair (Group of 2) Assignment



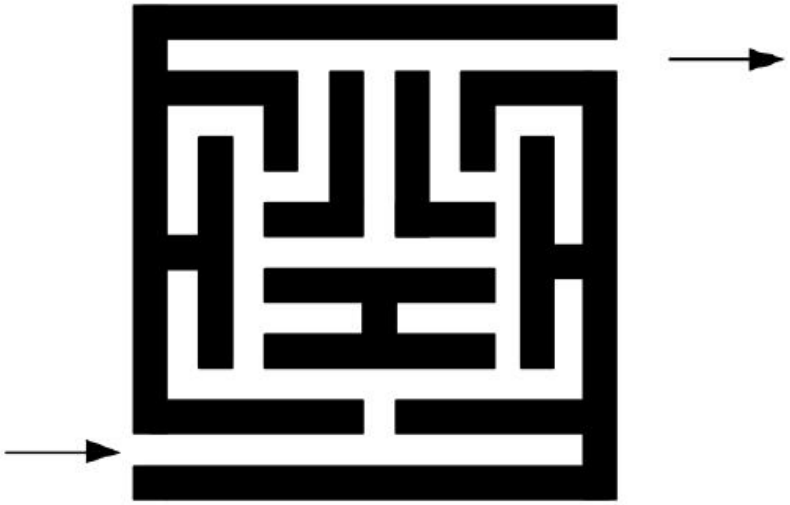




- Design and implement a number of algorithms to generate 2D mazes.
  - Recursive Backtracker (DFS) — Lecture 3
  - Prim's — Lecture 9
  - Growing Tree
- Design and implement a number of algorithms to solve these mazes.
  - Wall Follower — Brute force
  - Bidirectional Recursive Backtracker — Lecture 3, Tutorial 3

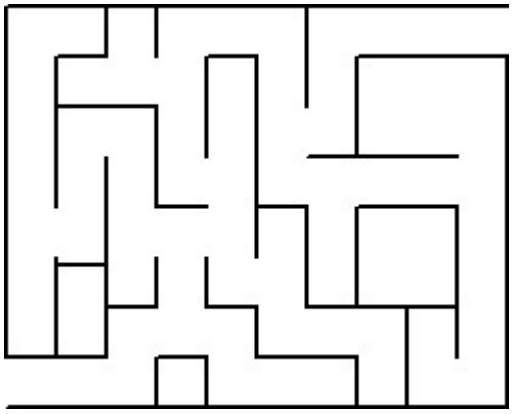
Tutorial 3

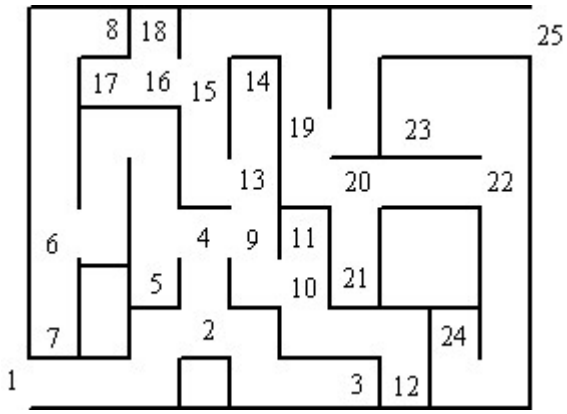
3.5.1 One can model a maze by having a vertex for a starting point, a finishing point, dead ends, and all the points in the maze where more than one path can be taken, and then connecting the vertices according to the paths in the maze.

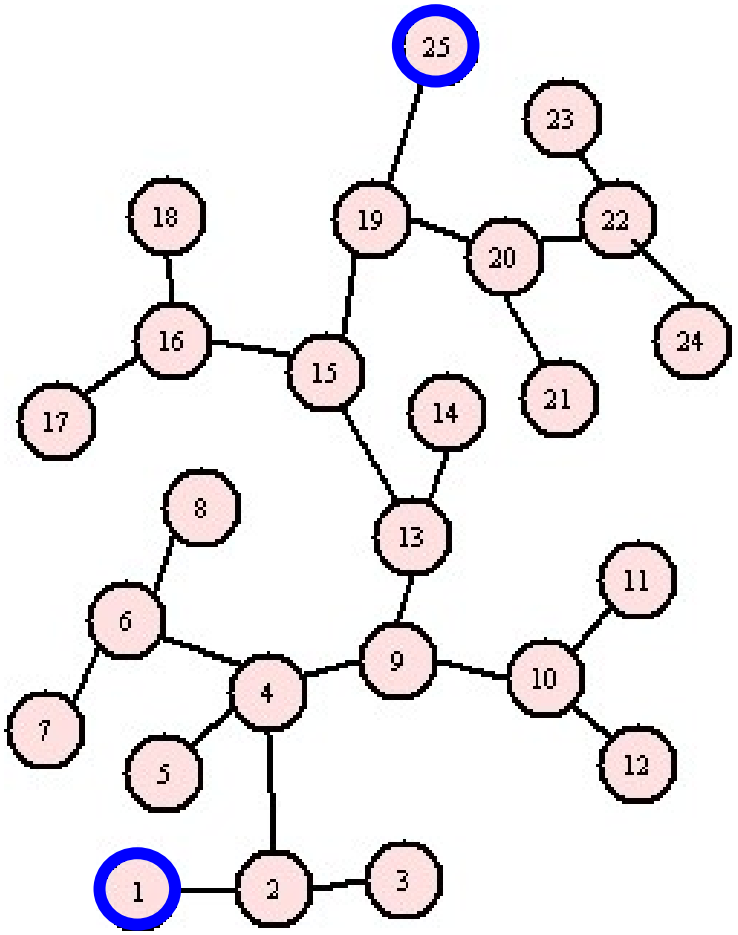


a Construct such a graph for the maze.

b Which traversal – DFS or BFS – would you use if you found yourself in a maze and why?







https://secweb.cs.odu.edu/~zeil/cs361/web/website/Lectures/graphalgorithms/pages/ar01s02.html

- There are two example parameter files in the skeleton code.
  - 'example1. para'
  - 'example2.para'.
- FAQ: Blackboard -> Assignments -> Assignment 2.
- **Note:**
  - isSolved() function in (MazeSolver.java / WallFollwerSolver.java / BiDirectionalRecursiveBacktrackerSolver.java) has designed to help you test your solvers. Specifically, for the isSolved() method, implement your own validation, but at bare minimum return true.
  - For validation (the validate() function in Maze.java) to work, **You must call drawFtPrt(cell), when you visit a cell in your solver (this is also necessary to visualise the solver).** This method is used to track the path your solver goes through and allows the new code to check.
- Remember I strongly suggested for no changes made to maze classes or MazeTester.

- **Note:** Make the visualization work on coreteaching servers
  - Xming + Putty (X11) for Windows
  - Xquartz + SSH -X for Mac/Linux

- Error:
  - WallFollowerSolver.java: Complete the implementation of this le.

Thanks!