For lab 6 you wrote a program that created a maze where each cell was reachable from any other cell and there was a unique path from the start to the destination. Your program works by removing one wall at a time (making sure that the cells separated by that wall were not reachable from each other) until the disjoint set forest representing the maze has exactly one tree. For this lab you will implement methods to find and display the path from the source to the destination in your maze.

1. Write a method to build the adjacency list representation of your maze. Cells in the maze should be represented by vertices in the graph. If two cells $u$ and $v$ are contiguous and there is no wall separating them, then there must be an edge from $u$ to $v$ in the graph. The example below shows a maze and the corresponding graph representation.

2. Implement the following algorithms to solve the maze you created. You may use the implementations of stacks and queues provided by Java or create your own.
   
   (a) Breadth-first search.
   
   (b) Depth-first search using a stack. This is identical to breadth-first search but the queue is replaced by a stack.
   
   (c) Depth-first search using recursion.

3. As usual, write a report describing your results. Display the paths found by your algorithms and compare their running times for different maze sizes.