## CS1020 Lab #5
## Exercise #1: Shortest Path Problem

http://www.comp.nus.edu.sg/~cs1020/3_ca/labs.html

**Learning Objectives:**

(1) Working with two-dimensional array.
(2) Writing recursive function.
(3) Use of LinkedList/Stack and Point.

**Task Statement:**

One common task in many computer games is to find the shortest path out of an *N* x *N* maze. In this lab, we will be representing the maze using a 2-dimensional integer array where 0 denotes a path and 1 denotes a wall. The minimum size of the maze is 3 x 3, the maximum size 20 x 20, and the entrance to the maze will always be located at (1,0). Rows are numbered from top to bottom (hence the top-most row is row 0), and columns are numbered from left to right (the left-most column is column 0). There may be multiple exits, which are located along the top, right, and bottom boundaries of the maze.

Write a program **ShortestPath.java** to read in the size of the square maze, followed by the grids containing the paths and walls. Your program will then compute and output the coordinates of the shortest path out of this maze. Your solution must make use of a recursive function and either a **LinkedList** or **Stack** data structure which holds a Java **Point** object.

You may assume that there are no cyclic paths, no path leads to a dead end, and that there is only 1 shortest path.

**Sample Input 1:**

```
10
1 1 1 0 1 1 1 1 1 1
0 0 1 0 1 1 1 1 1 1
1 0 1 0 0 1 1 1 1 1
1 0 1 1 0 1 1 1 1 1
1 0 0 0 0 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
```

**Sample Output 1:**

The shortage path is 13 steps:

(1, 0)
(1, 1)
(2, 1)
(3, 1)
(4, 1)
(4, 2)
(4, 3)
(4, 4)
(3, 4)
(2, 4)
(2, 3)
(1, 3)
(0, 3)

The shortest path is illustrated in bold red font below:

1 1 1 **0** 1 1 1 1 1 1
**0 0** 1 **0** 1 1 1 1 1 1
1 **0** 1 **0 0** 1 1 1 1 1
1 **0** 1 1 **0** 1 1 1 1 1
1 **0 0 0 0** 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1

**Sample Input 2:**

10
1 1 1 0 1 1 1 1 1 1
0 0 1 0 1 1 1 1 1 1
1 0 1 0 0 1 1 1 1 1
1 0 1 1 0 1 1 1 1 1
1 0 0 0 0 0 0 1 1 1
1 0 1 1 1 1 0 1 0 0
1 0 1 1 1 1 0 0 0 1
1 0 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 0 1 1 1

**Sample Output 2:**

The shortage path is 13 steps:

(1, 0)
(1, 1)
(2, 1)
(3, 1)
(4, 1)
(4, 2)
(4, 3)
(4, 4)
(3, 4)
(2, 4)
(2, 3)
(1, 3)
(0, 3)

The shortest path is illustrated in bold red font below:

1 1 1 **0** 1 1 1 1 1 1
**0 0** 1 **0** 1 1 1 1 1 1
1 **0** 1 **0 0** 1 1 1 1 1
1 **0** 1 1 **0** 1 1 1 1 1
1 **0 0 0 0** 0 0 1 1 1
1 0 1 1 1 1 0 1 0 0
1 0 1 1 1 1 0 0 0 1
1 0 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 0 0

**Sample Input 3:**

15
1 1 1 0 1 1 0 1 1 1 1 0 1 0 1
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 0 1 1 1 1 1 1 1 1 1 1
1 0 1 1 0 1 1 1 1 1 1 1 1 1 1
1 0 1 1 0 0 0 1 1 1 1 1 1 1 1
1 0 1 1 1 1 0 0 0 0 0 0 0 0 0
1 0 1 1 1 1 0 1 1 1 1 1 1 1 1
1 0 1 1 1 1 0 1 1 1 1 1 1 0 0
1 0 1 1 1 1 0 1 1 1 1 0 0 0 1
1 0 1 1 1 1 0 1 1 1 1 0 1 1 1
1 0 1 1 1 1 0 1 1 1 1 0 1 1 1
1 0 1 1 1 1 0 1 1 1 1 0 1 1 1
1 0 1 1 1 1 0 1 1 1 1 0 1 1 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

**Sample Output 3:**

The shortage path is 27 steps:
(1, 0)
(1, 1)
(2, 1)
(3, 1)
(4, 1)
(5, 1)
(6, 1)
(7, 1)
(8, 1)
(9, 1)
(10, 1)
(11, 1)
(12, 1)
(13, 1)
(13, 2)
(13, 3)
(13, 4)
(13, 5)
(13, 6)
(13, 7)
(13, 8)
(13, 9)
(13, 10)
(13, 11)
(13, 12)
(13, 13)
(13, 14)

The shortest path is illustrated in bold red font below:

1 1 1 0 1 1 0 1 1 1 1 0 1 0 1
**0 0** 1 0 0 0 0 0 0 0 0 0 0 0 0
1 **0** 1 1 0 1 1 1 1 1 1 1 1 1 1
1 **0** 1 1 0 1 1 1 1 1 1 1 1 1 1
1 **0** 1 1 0 0 0 1 1 1 1 1 1 1 1
1 **0** 1 1 1 1 0 0 0 0 0 0 0 0 0
1 **0** 1 1 1 1 0 1 1 1 1 1 1 1 1
1 **0** 1 1 1 1 0 1 1 1 1 1 1 0 0
1 **0** 1 1 1 1 0 1 1 1 1 0 0 0 1
1 **0** 1 1 1 1 0 1 1 1 1 0 1 1 1
1 **0** 1 1 1 1 0 1 1 1 1 0 1 1 1
1 **0** 1 1 1 1 0 1 1 1 1 0 1 1 1
1 **0** 1 1 1 1 0 1 1 1 1 0 1 1 1
1 **0 0 0 0 0 0 0 0 0 0 0 0 0 0**
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1