

Problem 1

Given a maze in the form of the binary rectangular matrix, find the shortest path's length in a maze from a given source to a given destination.

The path can only be constructed out of cells having value 1, and at any given moment, we can only move one step in one of the four directions (top, right, bottom, left).

Examples

For example, consider the following binary matrix. If source = (0, 0) and destination = (7, 5), the shortest path from source to destination has length 12 (a move from one cell to the next is an edge, so 13 cells means path size is 12).

```
[ 1 1 1 1 1 0 0 1 1 1 ]
[ 0 1 1 1 1 1 0 1 0 1 ]
[ 0 0 1 0 1 1 1 0 0 1 ]
[ 1 0 1 1 1 0 1 1 0 1 ]
[ 0 0 0 1 0 0 0 1 0 1 ]
[ 1 0 1 1 1 0 0 1 1 0 ]
[ 0 0 0 0 1 0 0 1 0 1 ]
[ 0 1 1 1 1 1 1 1 0 0 ]
[ 1 1 1 1 1 0 0 1 1 1 ]
[ 0 0 1 0 0 1 1 0 0 1 ]
```

Hints

- Create an empty queue and enqueue the source cell having a distance 0 from the source (itself) and mark it as visited.
- Loop till queue is empty.
 - Dequeue the front node.
 - If the popped node is the destination node, then return its distance.
 - Otherwise, for each of four adjacent cells of the current cell, enqueue each valid cell with +1 distance and mark them as visited.
- If all the queue nodes are processed, and the destination is not reached, then return false.

References

- <https://www.techiedelight.com/lee-algorithm-shortest-path-in-a-maze/>