Shortest Distance in a Binary Maze

Given a **n * m** matrix **grid** where each element can either be **0** or **1**. You need to find the shortest distance between a given source cell to a destination cell. The path can only be created out of a cell if its value is 1.

If the path is not possible between source cell and destination cell, then return **-1**.

Note : You can move into an adjacent cell if that adjacent cell is filled with element 1. Two cells are adjacent if they share a side. In other words, you can move in one of the four directions, Up, Down, Left and Right. The source and destination cell are based on the zero based indexing.

Example 1:

```
Input:
grid[][] = {{1, 1, 1, 1},
            {1, 1, 0, 1},
            {1, 1, 1, 1},
            {1, 1, 0, 0},
            {1, 0, 0, 1}}
source = {0, 1}
destination = {2, 2}
Output:
3
Explanation:
1 1 1 1
1 1 0 1
1 1 1 1
1 1 0 0
1 0 0 1
The highlighted part in the matrix denotes the
```

```
shortest path from source to destination cell.
```

Example 2:

```
Input:
grid[][] = {{1, 1, 1, 1, 1},
            {1, 1, 1, 1, 1},
            {1, 1, 1, 1, 0},
            {1, 0, 1, 0, 1}}
source = {0, 0}
destination = {3, 4}
Output:
-1
Explanation:
The path is not possible between source and
destination, hence return -1.
```

Your Task:

You don't need to read or print anything. Your task is to complete the function **shortestPath()** which takes the a 2D integer array **grid**, **source** cell and **destination** cell as an input parameters and returns the shortest distance between source and destination cell.

Expected Time Complexity: O(n * m)
Expected Space Complexity: O(n * m)

Constraints:

- $1 \leq n, m \leq 500$
- grid[i][j] == 0 or grid[i][j] == 1
- The source and destination cells are always inside the given matrix.