

Analyzing Time & Space Complexity – Spot Question

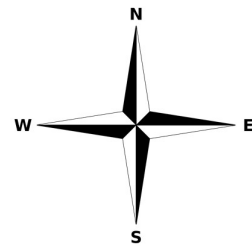
Consider a maze represented as a matrix of $m \times n$ with walls marked as 0 and a movable path marked as 1. From any point in the maze, the only movements possible are to the east or to the south. The exception to the movement is when a dead end is encountered, which is handled by backtracking to the previous position. Implement a function (findPath) to determine the only existing path from the north west corner to the south east corner of the maze. [Note: the path should be a sequence of directions: S / E]

Function Prototype: char * findPath(int **maze, int m, int n)

Sample:

input:

Start →	1	1	1	0
	1	0	1	0
	0	1	1	1
	0	0	1	0
	0	0	1	1 ← Destination



output: EESSSE