solve1.py

```python
# solve1.py
# define the maze as an array
# 2 is the start
# 3 is the end
# 1 is a path
# 0 is blocked

#10x10 list of lists
maze = [
    [2, 0, 3, 1, 1, 0, 1, 1, 1, 1],
    [1, 0, 1, 0, 1, 0, 0, 0, 0, 1],
    [1, 1, 1, 0, 1, 0, 0, 1, 1, 1],
    [1, 0, 1, 0, 0, 0, 0, 1, 0, 0],
    [1, 0, 1, 1, 1, 1, 1, 1, 0, 1],
    [1, 0, 0, 0, 0, 0, 0, 1, 0, 1],
    [1, 1, 1, 1, 0, 1, 0, 1, 0, 1],
    [1, 0, 0, 0, 0, 1, 0, 1, 1, 1],
    [1, 0, 1, 0, 0, 1, 0, 0, 0, 1],
    [1, 1, 1, 1, 1, 1, 0, 1, 1, 1]
]


def solve(maze):

    # get the number of rows and columns
    rows = len(maze)
    cols = len(maze[0]) if rows > 0 else 0

    # find the start position
    start = None
    for i in range(len(maze)):
        for j in range(len(maze[i])):
            if maze[i][j] == 2:
                start = (i, j)
                break
        # we found start so stop looking
        if start:
            break

    # find the end position
    end = None
    for i in range(len(maze)):
        for j in range(len(maze[i])):
            if maze[i][j] == 3:
                end = (i, j)
                break
        # we found end so stop looking
        if end:
            break

    if not start or not end:
```

```
        return None

    # initialize the path
    path = []

    #create and set visited to false
    visited = [[False]*cols for _ in range(rows)]

    def check(r, c):
        if not (0 <= r < rows and 0 <= c < cols):
            return False
        if maze[r][c] == 0 or visited[r][c]:
            return False

        visited[r][c] = True
        #print(f"Visiting: ({r}, {c})")

        path.append((r, c))

        if (r, c) == end:
            return True

        # Check all four directions recursively
        if (check(r+1, c) or check(r-1, c) or check(r, c+1) or check(r, c-1)):
            return True

        # If none of the directions lead to a solution, backtrack
        path.pop()
        return False




    if check(*start):
        return path
    else:
        return None

# Test
path = solve(maze)
if path:
    print("Path found:")
    for step in path:
        print(step)
else:
    print("No path found.")


```