```python
def dfs(maze, start, end):
    stack = [(start, [start])]  # Each element is (position, path_so_far)
    visited = set()

    while stack:
        position, path = stack.pop()
        x, y = position

        if position == end:
            return path  # Return the successful path

        visited.add(position)

        # Explore neighbors (up, down, left, right)
        for dx, dy in [(-1, 0), (1, 0), (0, -1), (0, 1)]:
            new_x, new_y = x + dx, y + dy
            new_pos = (new_x, new_y)

            # Check bounds, walls, and visited
            if (0 <= new_x < len(maze) and 0 <= new_y < len(maze[0]) and
                maze[new_x][new_y] == 0 and new_pos not in visited):
                stack.append((new_pos, path + [new_pos]))

    return None  # No path found


# Example maze: 0 -> open path, 1 -> wall
maze = [
    [0, 1, 0, 0, 0],
    [0, 1, 0, 1, 0],
    [0, 0, 0, 1, 0],
    [1, 1, 1, 1, 0],
    [0, 0, 0, 0, 0]
]

start = (0, 2)
end = (3, 4)

path = dfs(maze, start, end)
if path:
```

```
    print("Path found:", path)
else:
    print("No path exists")
```