

```

def dfs(maze, start, end):
    stack = [(start, [start])] # Stack stores tuples: (current_position, path_so_far)

    while stack:
        (x, y), path = stack.pop() # Pop the last element (LIFO)

        if (x, y) == end:
            return path # Return the path when we reach the end

        for dx, dy in [(1,0), (0,1), (-1,0), (0,-1)]: # Down, Right, Up, Left
            next_x, next_y = x + dx, y + dy
            if maze[next_x][next_y] != '#' and (next_x, next_y) not in path:
                stack.append(((next_x, next_y), path + [(next_x, next_y)]))

# Example usage
maze = [
    ['#', '#', '#', '#', '#', '#'],
    ['#', 'S', ' ', ' ', ' ', '#'],
    ['#', ' ', '#', ' ', '#', '#'],
    ['#', ' ', '#', ' ', ' ', '#'],
    ['#', ' ', ' ', ' ', 'E', '#'],
    ['#', '#', '#', '#', '#', '#']
]

start = (1, 1) # 'S' position
end = (4, 4) # 'E' position

path = dfs(maze, start, end)
print("DFS Path:", path)

```