```
from queue import Queue

def bfs(maze, start, end):
    queue = Queue()
    queue.put([start])  # Enqueue the start position

    while not queue.empty():
        path = queue.get()  # Dequeue the path
        x, y = path[-1]     # Current position is the last element of the path

        if (x, y) == end:
            return path  # Return the path if end is reached

        for dx, dy in [(1,0), (0,1), (-1,0), (0,-1)]:  # Possible movements
            next_x, next_y = x + dx, y + dy
            if maze[next_x][next_y] != '#' and (next_x, next_y) not in path:
                new_path = list(path)
                new_path.append((next_x, next_y))
                queue.put(new_path)  # Enqueue the new path

# Example usage
maze = [
    ['#', '#', '#', '#', '#', '#'],
    ['#', 'S', ' ', ' ', ' ', '#'],
    ['#', ' ', '#', ' ', '#', '#'],
    ['#', ' ', '#', ' ', ' ', '#'],
    ['#', ' ', ' ', ' ', 'E', '#'],
    ['#', '#', '#', '#', '#', '#']
]
start = (1, 1)  # Start position (S)
end = (4, 4)    # End position (E)
path = bfs(maze, start, end)
print(path)
```