def puzzle\_maze(maze, start, goal):

queue = []

# Using a list instead of deque

visited = set()

parent = {}

queue.append(start)

visited.add(start)

while queue:

current = queue.pop(0)

# Pop from the front (simulates a queue)

if current == goal:

# Path found, backtrack to get the path

path = []

while current != start:

path.append(current)

current = parent[current]

path.append(start)

path.reverse()

return path

# Check adjacent nodes

for action in [(0, 1), (0, -1), (1, 0), (-1, 0)]:

next\_node = (current[0] + action[0], current[1] + action[1])

if next\_node in maze and next\_node not in visited:

queue.append(next\_node)

visited.add(next\_node)

parent[next\_node] = current

return None

maze = {(0, 0), (0, 1), (0, 2), (1, 2), (2, 2), (2, 3), (3, 3)}

start = (0, 0)

goal = (3, 3)

path = puzzle\_maze(maze, start, goal)

if path:

print("Path found:", path)

else:

print("No path found")