Exercise 2

from collections import deque

def water\_jug\_bfs(jug1\_capacity, jug2\_capacity, goal):

visited = set()

queue = deque()

# Each state is (jug1, jug2)

queue.append((0, 0))

while queue:

jug1, jug2 = queue.popleft()

if (jug1, jug2) in visited:

continue

print(f"Jug1: {jug1}L, Jug2: {jug2}L")

visited.add((jug1, jug2))

# Check if goal is reached

if jug1 == goal or jug2 == goal:

print("Goal reached!")

return True

# Possible states

states = [

(jug1\_capacity, jug2), # Fill Jug1

(jug1, jug2\_capacity), # Fill Jug2

(0, jug2), # Empty Jug1

(jug1, 0), # Empty Jug2

# Pour Jug1 -> Jug2

(jug1 - min(jug1, jug2\_capacity - jug2),

jug2 + min(jug1, jug2\_capacity - jug2)),

# Pour Jug2 -> Jug1

(jug1 + min(jug2, jug1\_capacity - jug1),

jug2 - min(jug2, jug1\_capacity - jug1)),

]

for state in states:

if state not in visited:

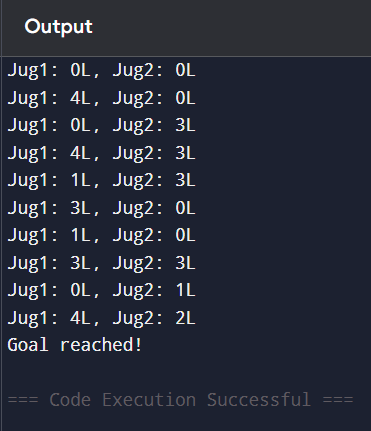
queue.append(state)

print("No solution.")

return False

# Example usage

water\_jug\_bfs(4, 3, 2)

S