from collections import deque

def is\_visited(state, visited):

return state in visited

def water\_jug\_solver(jug1\_capacity, jug2\_capacity, target):

visited = set()

queue = deque([(0, 0)])

while queue:

jug1, jug2 = queue.popleft()

if (jug1, jug2) in visited:

continue

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

visited.add((jug1, jug2))

if jug1 == target or jug2 == target:

print("Solution found!")

return

# Fill Jug1

queue.append((jug1\_capacity, jug2))

# Fill Jug2

queue.append((jug1, jug2\_capacity))

# Empty Jug1

queue.append((0, jug2))

# Empty Jug2

queue.append((jug1, 0))

# Pour Jug1 to Jug2

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

queue.append((jug1 - pour, jug2 + pour))

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

queue.append((jug1 + pour, jug2 - pour))

print("No solution found")

# Test the solver function

jug1\_capacity = 4

jug2\_capacity = 3

target = 2

water\_jug\_solver(jug1\_capacity, jug2\_capacity, target)

Output:

