from collections import deque

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

visited = set()

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

while queue:

jug1, jug2 = queue.popleft()

if jug1 == target or jug2 == target:

return (jug1, jug2)

# Fill Jug 1

if jug1 < jug1\_capacity:

fill\_jug1 = (jug1\_capacity, jug2)

if fill\_jug1 not in visited:

visited.add(fill\_jug1)

queue.append(fill\_jug1)

# Fill Jug 2

if jug2 < jug2\_capacity:

fill\_jug2 = (jug1, jug2\_capacity)

if fill\_jug2 not in visited:

visited.add(fill\_jug2)

queue.append(fill\_jug2)

# Empty Jug 1

if jug1 > 0:

empty\_jug1 = (0, jug2)

if empty\_jug1 not in visited:

visited.add(empty\_jug1)

queue.append(empty\_jug1)

# Empty Jug 2

if jug2 > 0:

empty\_jug2 = (jug1, 0)

if empty\_jug2 not in visited:

visited.add(empty\_jug2)

queue.append(empty\_jug2)

# Pour from Jug 1 to Jug 2

if jug1 > 0 and jug2 < jug2\_capacity:

pour\_jug1\_to\_jug2 = (max(jug1 - (jug2\_capacity - jug2), 0), min(jug2 + jug1, jug2\_capacity))

if pour\_jug1\_to\_jug2 not in visited:

visited.add(pour\_jug1\_to\_jug2)

queue.append(pour\_jug1\_to\_jug2)

# Pour from Jug 2 to Jug 1

if jug2 > 0 and jug1 < jug1\_capacity:

pour\_jug2\_to\_jug1 = (min(jug1 + jug2, jug1\_capacity), max(jug2 - (jug1\_capacity - jug1), 0))

if pour\_jug2\_to\_jug1 not in visited:

visited.add(pour\_jug2\_to\_jug1)

queue.append(pour\_jug2\_to\_jug1)

return None

# Example usage:

jug1\_capacity = 4

jug2\_capacity = 3

target\_amount = 2

result = water\_jug\_bfs(jug1\_capacity, jug2\_capacity, target\_amount)

if result:

print("Solution found:")

print(f"Jug 1: {result[0]}, Jug 2: {result[1]}")

else:

print("No solution found.")