# Water Jug Problem Solver Using BFS (Python)

## Python Code:

from collections import deque  
  
def water\_jug\_bfs(jug1\_cap, jug2\_cap, target):  
 visited = set()  
 queue = deque()  
  
 queue.append(((0, 0), []))  
  
 while queue:  
 (a, b), path = queue.popleft()  
  
 if (a, b) in visited:  
 continue  
 visited.add((a, b))  
  
 path = path + [(a, b)]  
  
 if a == target or b == target:  
 print("Solution found!\nSteps:")  
 for step in path:  
 print(f"Jug1: {step[0]} gallons, Jug2: {step[1]} gallons")  
 return  
  
 next\_states = [  
 (jug1\_cap, b),  
 (a, jug2\_cap),  
 (0, b),  
 (a, 0),  
 (0, a + b) if a + b <= jug2\_cap else (a - (jug2\_cap - b), jug2\_cap),  
 (a + b, 0) if a + b <= jug1\_cap else (jug1\_cap, b - (jug1\_cap - a))  
 ]  
  
 for state in next\_states:  
 if state not in visited:  
 queue.append((state, path))  
  
 print("No solution found.")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 jug1\_capacity = 4  
 jug2\_capacity = 3  
 target\_amount = 2  
  
 water\_jug\_bfs(jug1\_capacity, jug2\_capacity, target\_amount)

## Sample Output:

Solution found!  
Steps:  
Jug1: 0 gallons, Jug2: 0 gallons  
Jug1: 0 gallons, Jug2: 3 gallons  
Jug1: 3 gallons, Jug2: 0 gallons  
Jug1: 3 gallons, Jug2: 3 gallons  
Jug1: 4 gallons, Jug2: 2 gallons  
Jug1: 0 gallons, Jug2: 2 gallons  
Jug1: 2 gallons, Jug2: 0 gallons