```python
from collections import deque

def water_jug_bfs(jug1, jug2, target):
    visited = set()
    queue = deque()

    # Initial state (0, 0)
    queue.append((0, 0))

    while queue:
        a, b = queue.popleft()

        # If this state is already visited, skip
        if (a, b) in visited:
            continue
        visited.add((a, b))

        print(f"Jug1: {a}, Jug2: {b}")

        # If target reached
        if a == target or b == target:
            print("\nTarget achieved!")
            return True

        # All possible next states
        next_states = [
            (jug1, b),           # Fill jug1
            (a, jug2),           # Fill jug2
            (0, b),              # Empty jug1
            (a, 0),              # Empty jug2
            (a - min(a, jug2 - b), b + min(a, jug2 - b)),  # Pour jug1 -> jug2
            (a + min(b, jug1 - a), b - min(b, jug1 - a))   # Pour jug2 -> jug1
        ]

        for state in next_states:
            if state not in visited:
                queue.append(state)

    print("\nNo solution possible.")
    return False


# Example usage
jug1 = 4   # capacity of Jug 1
jug2 = 3   # capacity of Jug 2
target = 2 # target amount of water
```

```
Jug1: 0, Jug2: 0
Jug1: 4, Jug2: 0
Jug1: 0, Jug2: 3
Jug1: 4, Jug2: 3
Jug1: 1, Jug2: 3
Jug1: 3, Jug2: 0
Jug1: 1, Jug2: 0
Jug1: 3, Jug2: 3
Jug1: 0, Jug2: 1
Jug1: 4, Jug2: 2

Target achieved!
```