# Name: Shantanu Rohile

# Roll No: 53

*# ==============================================*

# AI Practical 02-B

# Problem Statement:

# Write a program to solve the Water Jug Problem using Breadth-First Search (BFS) # ==============================================

from queue import Queue #

# BFS algorithm to solve the Water Jug Problem

*#*

def water\_jug\_bfs(capacity\_jug1, capacity\_jug2, target): visited\_states = set()

q = Queue()

q.put((0, 0)) # Starting state: both jugs are empty

while not q.empty():

current\_state = q.get()

if current\_state in visited\_states:

continue

visited\_states.add(current\_state) jug1, jug2 = current\_state

# Goal check: either jug has the target amount if jug1 == target or jug2 == target:

return current\_state

# Possible operations:

# 1. Fill Jug1 q.put((capacity\_jug1, jug2)) # 2. Fill Jug2

q.put((jug1, capacity\_jug2)) # 3. Empty Jug1

q.put((0, jug2)) # 4. Empty Jug2 q.put((jug1, 0))

# 5. Pour Jug1 → Jug2

pour\_amount = min(jug1, capacity\_jug2 - jug2) q.put((jug1 - pour\_amount, jug2 + pour\_amount)) # 6. Pour Jug2 → Jug1

pour\_amount = min(jug2, capacity\_jug1 - jug1) q.put((jug1 + pour\_amount, jug2 - pour\_amount))

return None # No solution found

*#*

# Example Usage #

capacity\_jug1 = 5

capacity\_jug2 = 3

target = 4,0

result = water\_jug\_bfs(capacity\_jug1, capacity\_jug2, target) print("Water Jug Solution:", result)

*#*

# Sample Output:

# Water Jug Solution: (4, 0) #