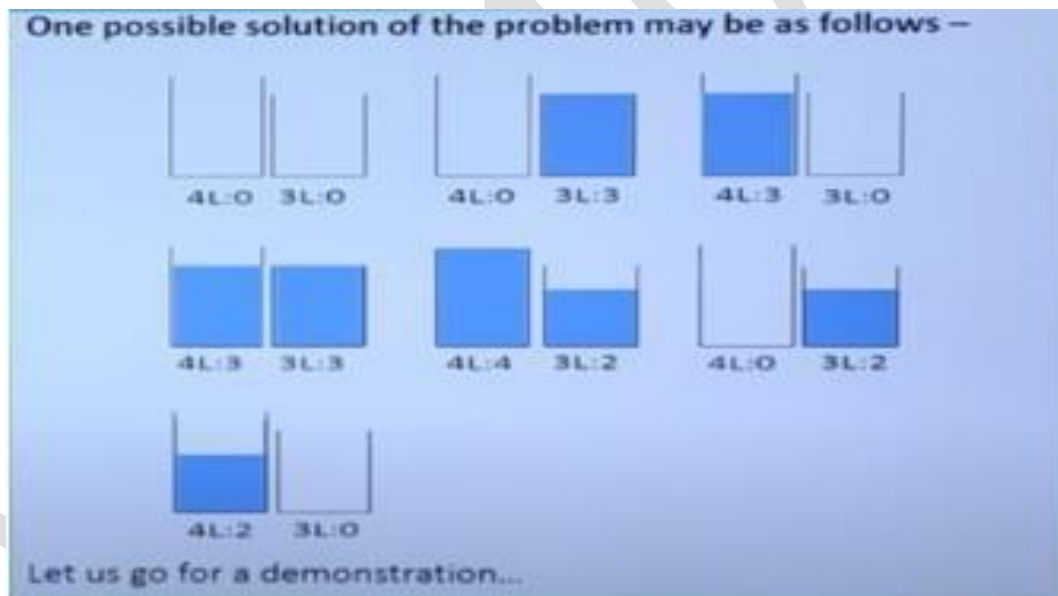


EX.NO: 03

DATE:

DEPTH-FIRST SEARCH – WATER JUG PROBLEM

In the **water jug problem in Artificial Intelligence**, we are provided with two jugs: one having the capacity to hold 3 gallons of water and the other has the capacity to hold 4 gallons of water. There is no other measuring equipment available and the jugs also do not have any kind of marking on them. So, the agent's task here is to fill the 4-gallon jug with 2 gallons of water by using only these two jugs and no other material. Initially, both our jugs are empty.



**CODE:**

```

def is_valid_state(state, visited):
    ⚡ return state not in visited

def dfs(current_state, target, jug1_capacity, jug2_capacity, visited, solution):
    visited.add(current_state)
    solution.append(current_state)

    if current_state[0] == target or current_state[1] == target:
        return True

    jug1, jug2 = current_state

    possible_states = [
        (jug1_capacity, jug2),
        (jug1, jug2_capacity),
        (0, jug2),
        (jug1, 0),
        (max(jug1 - (jug2_capacity - jug2), 0), min(jug2 + jug1, jug2_capacity)),
        (min(jug1 + jug2, jug1_capacity), max(jug2 - (jug1_capacity - jug1), 0)),
    ]

    for state in possible_states:
        if is_valid_state(state, visited):
            if dfs(state, target, jug1_capacity, jug2_capacity, visited, solution):
                return True

    solution.pop()
    return False

def water_jug_problem(jug1_capacity, jug2_capacity, target):
    visited = set()
    solution = []

    if dfs((0, 0), target, jug1_capacity, jug2_capacity, visited, solution):
        return solution
    else:
        return "No solution found."

jug1_capacity = 4
jug2_capacity = 3
target = 2

solution = water_jug_problem(jug1_capacity, jug2_capacity, target)
print("Solution steps:")
for step in solution:
    print(step)

```

**OUTPUT:**

Solution steps:

(0, 0)

(4, 0)

(4, 3)

(0, 3)

(3, 0)

(3, 3)

(4, 2)

**RESULT:**

Thus, the water jug program has been implemented successfully.