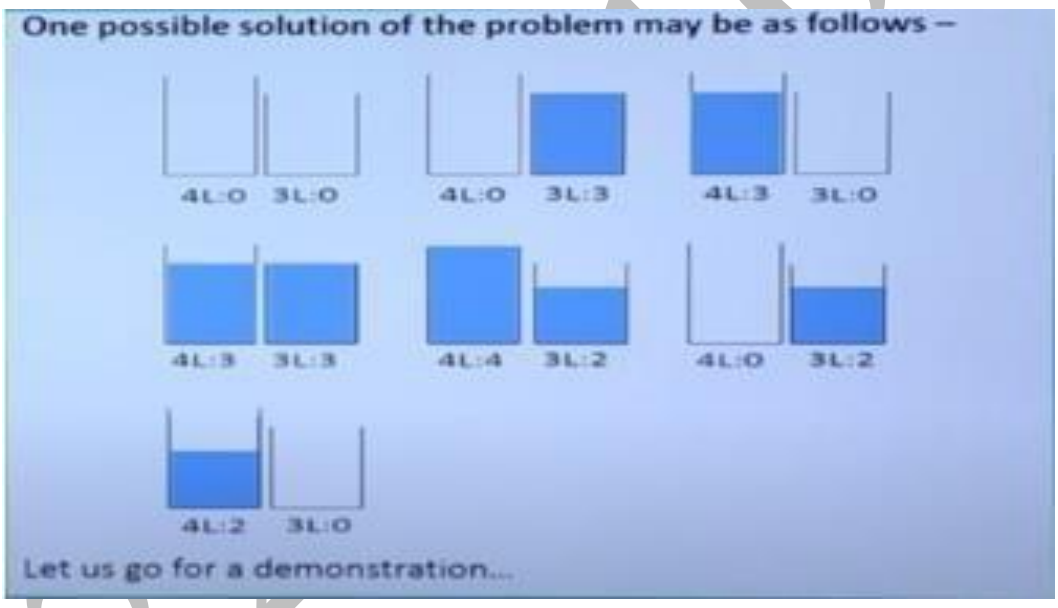
EX.NO:3 **DATE:13/9/2024**
Reg.no:220701060

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.

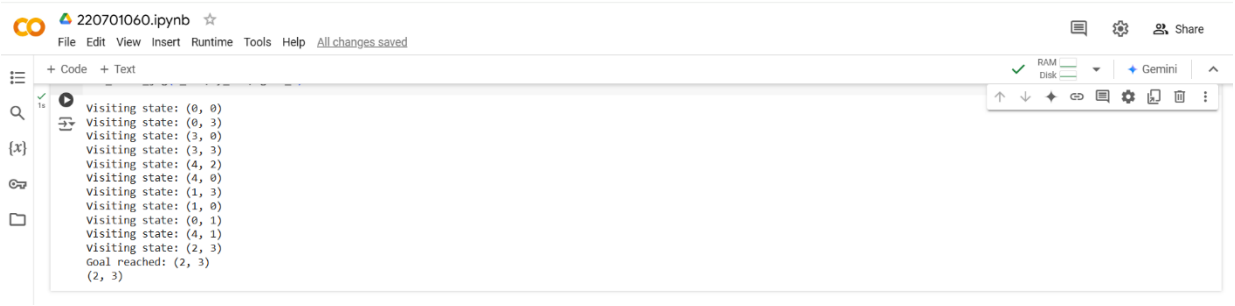


CODE:

```
def fill_3_gallon(x, y, x_max, y_max):
    return (x, y_max)

def empty_4_gallon(x, y, x_max, y_max):
    return (0, y)

def empty_3_gallon(x, y, x_max, y_max):
    return (x, 0)

def pour_4_to_3(x, y, x_max, y_max):
    transfer = min(x, y_max - y)
    return (x - transfer, y + transfer)

def pour_3_to_4(x, y, x_max, y_max):
    transfer = min(y, x_max - x)
    return (x + transfer, y - transfer)

def dfs_water_jug(x_max, y_max, goal_x, visited=None, start=(0, 0)):
    if visited is None:
        visited = set()
    stack = [start]

    while stack:
        state = stack.pop()
        x, y = state

        # Skip if already visited
        if state in visited:
            continue

        # Mark as visited
        visited.add(state)
        print(f"Visiting state: {state}")
```

```python
        # Check if goal is reached
        if x == goal_x:
            print(f"Goal reached: {state}")
            return state

        # Generate next possible states
        next_states = [
            fill_4_gallon(x, y, x_max, y_max),
            fill_3_gallon(x, y, x_max, y_max),
            empty_4_gallon(x, y, x_max, y_max),
            empty_3_gallon(x, y, x_max, y_max),
            pour_4_to_3(x, y, x_max, y_max),
            pour_3_to_4(x, y, x_max, y_max)
        ]

        # Add unvisited states to stack
        for new_state in next_states:
            if new_state not in visited:
                stack.append(new_state)

    print("No solution found")
    return None

# Problem definition
x_max = 4  # Maximum capacity of 4-gallon jug
y_max = 3  # Maximum capacity of 3-gallon jug
goal_x = 2 # Desired amount in the 4-gallon jug

# Solve the problem
dfs_water_jug(x_max, y_max, goal_x)
```

OUTPUT:



RESULT:

Thus, the water jug program has been executed successfully.