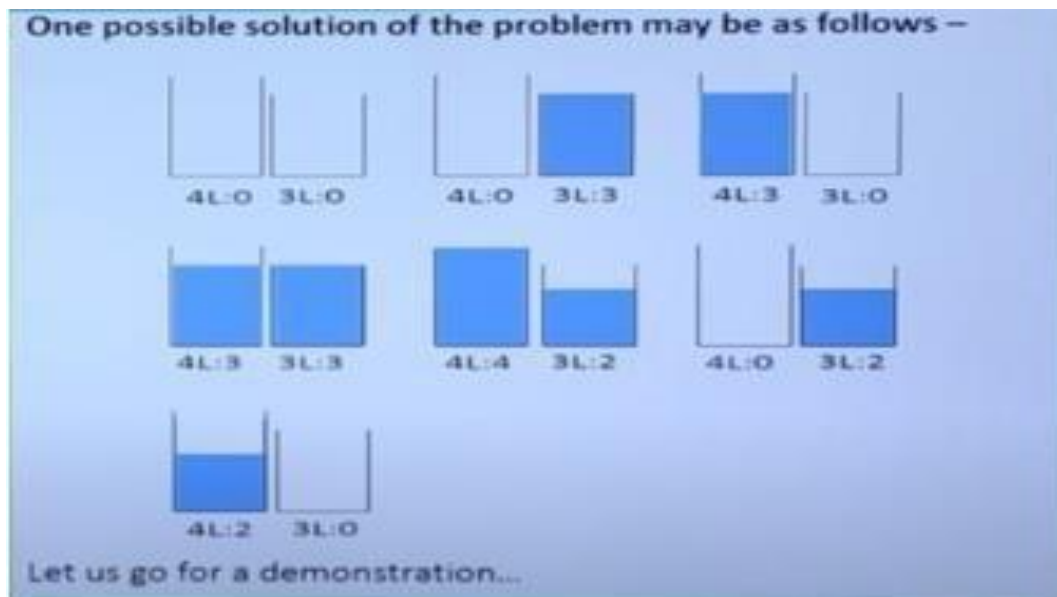


EX.NO: 03**DEPTH-FIRST SEARCH – WATER JUG PROBLEM**

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.

**AIM:**

To implement a python program for Water Jug problem using depth first search problem.

CODE:

```

from collections import deque

def DFS(a, b, target):
    m, path = {}, []
    q = deque([(0, 0)])

    while q:
        u = q.popleft()
        if (u[0], u[1]) in m or u[0] < 0 or u[1] < 0 or u[0] > a or u[1] > b:
            continue

        path.append([u[0], u[1]])
        m[(u[0], u[1])] = 1

        if u[0] == target or u[1] == target:
            if u[0] == target and u[1] != 0:
                path.append([u[0], 0])
            elif u[1] == target and u[0] != 0:
                path.append([0, u[1]])
            for p in path:
                print(f"({p[0]}, {p[1]})")
            return

        q.extend([(u[0], b), (a, u[1])])

        for ap in range(max(a, b) + 1):
            if (c := u[0] + ap) <= a and (d := u[1] - ap) >= 0:
                q.append([c, d])
            if (c := u[0] - ap) >= 0 and (d := u[1] + ap) <= b:
                q.append([c, d])

    print("No solution")

Jug1, Jug2, target = 4, 3, 2
print("Path from initial state to solution state:")
DFS(Jug1, Jug2, target)

```

OUTPUT:

```
⇒ Path from initial state to solution state:  
  (0, 0)  
  (0, 3)  
  (4, 0)  
  (4, 3)  
  (1, 2)  
  (0, 2)
```

RESULT:

Thus, the implementing of Water Jug problem using depth first search problem is successfully executed and the output is verified.