**EX.NO:3 DATE:16/10/2024**

**Reg.no:220701048**

**DEPTH-FIRST SEARCH – WATER JUG PROBLEM**

**AIM**:To implement water jug problem using DFS

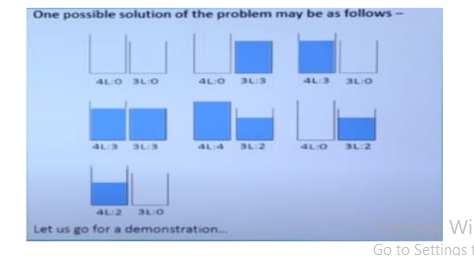
In the water jug problem in Artificial Intelligence, we are provided with two jugs: one having

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\_4\_gallon(x, y, x\_max, y\_max):

return (x\_max, y)

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) # Max amount we can transfer from 4-gallon to 3-gallon jug

return (x - transfer, y + transfer)

def pour\_3\_to\_4(x, y, x\_max, y\_max):

transfer = min(y, x\_max - x) # Max amount we can transfer from 3-gallon to 4-gallon jug

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() # Set to keep track of visited states

stack = [start] # Stack to store the states for DFS traversal

while stack:

state = stack.pop()

x, y = state

if state in visited:

continue

visited.add(state)

print(f"Visiting state: {state}")

if x == goal\_x:

print(f"Goal reached: {state}")

return state

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)

]

for new\_state in next\_states:

if new\_state not in visited:

stack.append(new\_state)

return None

x\_max = 4

y\_max = 3

goal\_x = 2

dfs\_water\_jug(x\_max, y\_max, goal\_x)

**OUTPUT**:  
