Exp: 5

Water Jug Using DFS

```
def Solve Water jug problem (capacity-jug1, capacity jug2,
                             desired-quantity):
    Steck=[]
    Steck.append(0,0)
While Stack:
    current-state = steck.pop()
if current_state[0]==desired_quantity or
    current-state[1]== desired_quantity;
    return current.state.
next.States = generate Next States(current_state
 Capacity-jug1, Capacity-jug2)
Stack.extend(next_states)
return "no solution found"}

def generate Next States (state, capacity-jug1,
                          capacity_jug2):
    next_states = []
    next_states.append [(capacity-jug1, capacity jug2))
    next_states.append [(state[0], capacity-jug2))
    next_states.append ((0, state[1]))
    next_states.append ((state[0], 0))
Pour-amount = min(State[0], capacityjug2, State[1])
 next_State.append ((State[0]- Pour-amount, State[1]
Pour_amount = min(  + pour_amount;
    next_state.append((State[0]+ pour_amount, State[1]-
                                                Pour_amount
```

```
return next_states

solution = solve_water_jug_problem(4, 3, 2)
print("Solution:", solution)
```

Result:

The program was successfully executed and output is verified.