Exp. No :   DFS with Water Jug
Date :

AIM :

To implement the DFS algorithm to Water Jug, to measure exactly a target amount of water using two jugs with given capacities.

ALGORITHM :

Step - 1 : Intialise starting state with both jugs empty (0, and create a set visited to track.

Step - 2 : Mark current state as visited.

Step - 3 : If amount of water in either jug equals the target, print the solution and terminate

Step - 4 : Generate all possible next states.

Step - 5 : Recursively apply DFS to each unvisited state.

Step - 6 : If a solution found, terminate recursion

Step - 7 : If no solution, print no solution.

CODE :

```
def waterJugDFS(jug1_capacity, jug2_capacity, target, current_
                        state = None, visited = None):
    if visited is None:
        visited = set()
    if current_state is None:
        current_state = (0,0)
```

```
if current_state in visited:
    return False
visited.add (current_state)
jug1, jug2 = current_state
Print (f" Jug1 : {jug1}, Jug2 : {jug2}")
if jug1 == target or jug2 == target :
   Print (" solution found.")
   return True
Possible_moves = [ (jug1-capacity , jug2), (jug1, jug2-capacity),
                   (0, jug2) , (jug1, 0),
                   (min (jug1-capacity , jug1 + jug2),
                    max (jug2-capacity, jug1))),
                   (max(0, jug1 - (jug2-capacity - jug2)),
                    min (jug2-capacity, jug1 + jug2)) ]
for next_state in possible_moves:
    if waterJugDFS (jug1_capacity, jug2_capacity, target,
                    next_state, visited):
          return True
return False
if __name__ == "__main__":
   jug1_capacity = 4
   jug2_capacity = 3
   target = 2
   Print (" DFS Traversal for water Jug Problem:")
   if not waterJugDFS (jug1_capacity, jug2_capacity, target):
       Print ("No solution found.")
```

OutPUT :

```
DFS Traversal for water Jug Problem:
Jug1:0 , Jug2:0
Jug1:4 , Jug2:0
Jug1:4 , Jug2:3
Jug1:0 , Jug2:3
Jug1:3 , Jug2:0
Jug1:3 , Jug2:3
Jug1:4 , Jug2:2
Solution found!
```

RESULT :

The program has been successfully executed and the output is verified.