****

**Superior University Lahore**

**Programming for AI (Lab)**

**Lab Task 1**

**Water\_Jug Problem (Dynamic)**

**BS in Artificial Intelligence**

*Department of Software Engineering*

*Faculty of Computer Science & Information Technology*

*The Superior University, Lahore*

**Submitted by:**

|  |  |
| --- | --- |
| **Student Name:** | **Alisha Kanwal** |
| **Roll Number** | **Su92-bsaim-f23-105** |
| **Section** | **BSAI-4B** |

**Report**

# **Introduction:**

## Problem Statement:

The Water Jug Problem is a classic artificial intelligence and problem-solving challenge that involves measuring a specific quantity of water using two jugs of fixed capacities. This report details the implementation of a Water Jug Problem Solver using the Depth-First Search (DFS) algorithm in Python.

## Problem Understanding:

Given two jugs of known capacities, the objective is to determine a sequence of actions that will measure exactly a target amount of water. The allowed operations are:

* Filling a jug completely.
* Emptying a jug.

# **Methodology**:

To solve the problem, a DFS-based approach is used:

* The algorithm starts with both jugs empty (0,0).
* It explores all possible valid states using a stack for DFS traversal.
* A visited set is maintained to avoid redundant computations.
* The algorithm continues searching until the target amount of water is reached in either of the jugs.
* If a solution is found, the sequence of steps is printed; otherwise, a message indicating no solution is displayed.

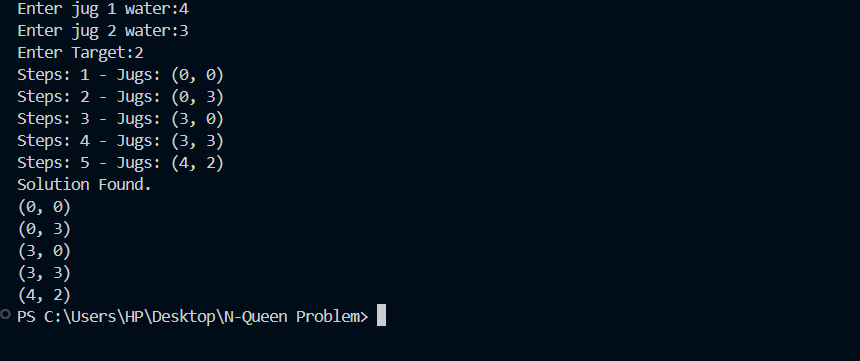
# **Implementation Details:**



The implementation in Python follows these steps:

* Accept user inputs for jug capacities and the target amount.
* Use a stack for DFS and a set to track visited states.
* Explore all possible transitions, including:
  + Filling a jug.
  + Emptying a jug.
  + Pouring from one jug to the other.
* Print each step taken in the solution path.
* Stop when the target amount is achieved in either jug.
* Output the final sequence of steps.

1. **Example Execution Input:**

****

# **Conclusion:**

This DFS-based Water Jug Problem Solver efficiently finds a solution when one exists by exploring all possible moves in a depth-first manner. The implementation ensures that redundant paths are not revisited and provides an intuitive step-by-step solution sequence. This problem-solving approach demonstrates the practical application of search algorithms in constraint-based problems.