Department of Computer Engineering

Academic Term II : 22-23

Class: T.E (Comp A), SemVI Subject Name: Artificial Intelligence

Student Name : Vailantan Fernandes Roll No: 9197

Practical No:	**2**
Title:	**Water Jug Problem Solving by using production system approach**
Date of Performance:	
Date of Submission:	

Rubrics for Evaluation:

Sr. No	Performance Indicator	Excellent	Good	Below Average	Marks
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Logic/Algorithm Complexity analysis(03)	03(Correct)	02(Partial)	01 (Tried)	
3	Coding Standards (03): Comments/indention/Naming conventions Test Cases /Output	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Assignment (03)	03(done well)	2 (Partially Correct)	1(submitted)	
Total					

Signature of the Teacher :

Experiment No: 2

Title: Water Jug Problem solving by using production system approach

Objective:To write a program that solves the water jug problem in an efficient manner

Theory: Given two unmarked jugs having capacities 'a' and 'b' liters respectively and a target volume't' liters, find the moves that get exactly 't' liters in any of the two jugs.

The problem is solvable only when t is a multiple of gcd(a,b) and can be modeled as search through a state space. The state space for this problem can be described as the set of ordered pair of integers (x,y) such that $x \in \{0,1,2,\ldots,a\}$ and $y \in \{0,1,2,\ldots,b\}$. The initial state is (0,0) and the goal states are (t,y) and (x,t)$\forall x,y$.

Algorithm:

1. All that isneeded for a search procedure to work is a way to generate new states (successors) from a given state.
2. This is captured by production rules that specify how and when a new state can be generated from a

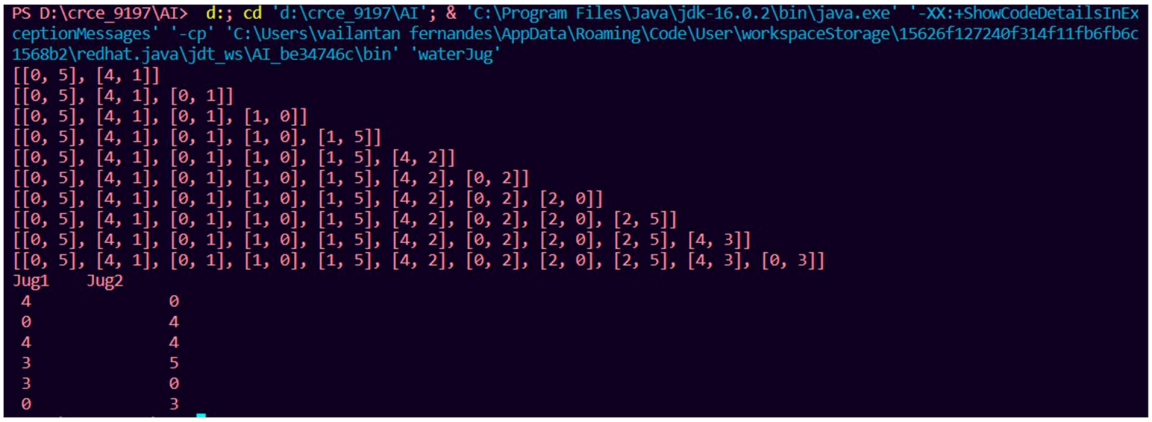
given state. For the water jug problem, the following production rules are

sufficient:

- (x, y) -> (a, y) if x < a i.e., Fill the first jug if it is not already full
- (x, y) -> (x, b) if y < b i.e., Fill the second jug if it is not already full
- (x, y) -> (0, y) if x > 0 i.e., Empty the first jug
- (x, y) -> (x, 0) if y > 0 i.e., Empty the second jug
- (x, y) -> (min(x + y, a), max(0, x + y – a)) if y > 0 i.e., Pour from second jug into first jug until the first jug is full or the second jug is empty
- (x, y) -> (max(0, x + y – b), min(x + y, b)) if x > 0 i.e., Pour from first jug into second jug until the second jug is full or the first jug is empty

3. Now, a search procedure like BFS or DFS can be applied to systematically search from the initial state to one of the goal states through the state space.

Output :



Post Lab Assignment:

1. I have three jug whose capacity are 8 lt.,5 lt. and 3 lt. and no one jug is calibrated then how can I divide 8 lt. water in two equal parts.
2. Write productions rules for water Jug Problem

   Enumerate depth and breadth first graph traversalsapplications.
3. Define the state space for water jug problem, travelingsalesman problem and 8-puzzle problem