***“*Heaven’s Light is Our Guide”**

**A logo with text and symbols

AI-generated content may be incorrect.**

**Rajshahi University of Engineering & Technology**

**Department of Computer Science & Engineering**

**Final Lab Report**

**Course Code: CSE 3208**

**Course Title: Artificial Intelligence Sessional**

|  |  |
| --- | --- |
| **Submitted By-**  **Name: Sajidur Rahman Tarafder**  **Department: CSE**  **Roll No: 2003154**  **Section: C**  **Session: 2020-21** | **Submitted To-**  **Md. Nasif Osman Khansur**  **Lecturer**  **Department of CSE,**  **RUET** |

**Title:** Solving the Water Jug Problem Using Rule-Based and Searching Algorithms (BFS, DFS) in Python.

**Introduction:** The Water Jug Problem is a well-known AI challenge used to demonstrate reasoning using rule-based systems and searching algorithms like BFS and DFS. Rule-based systems help define valid state transitions, while BFS and DFS help navigate the state space. Breadth-First Search (BFS) explores states level-wise and guarantees the shortest solution. Depth-First Search (DFS) explores one path deep before backtracking, which might not yield the shortest path but may be faster in some cases.

**Task-1:**

Solve the (4, 3) L Water Jug Problem using Traditional Rule Based System in Python.

**Rule-Based System:**

A rule-based system is an AI approach where the problem is solved using a fixed set of condition-action rules. Each rule defines a possible action (like filling, pouring, or emptying a jug) that transitions the system from one state to another. The system applies these rules sequentially to reach the desired goal state.

**Rules Used:**

1. Fill Jug1
2. Fill Jug2
3. Empty Jug1
4. Empty Jug2
5. Pour Jug2 into Jug1 until Jug1 is full
6. Pour Jug1 into Jug2 until Jug2 is full
7. Pour all from Jug2 to Jug1 if it fits
8. Pour all from Jug1 to Jug2 if it fits

**Code (Solution using Rule-Based System):**

|  |
| --- |
|  |
| **A computer screen with text  AI-generated content may be incorrect.** |

**Terminal Output:**

|  |
| --- |
|  |

**Task-2:**

Write Python code to determine how BFS and/or DFS can be used to solve the Water Jug Problem to reach to the state (2,x) or (x, 2) L.

**Code (Solution using BFS):**

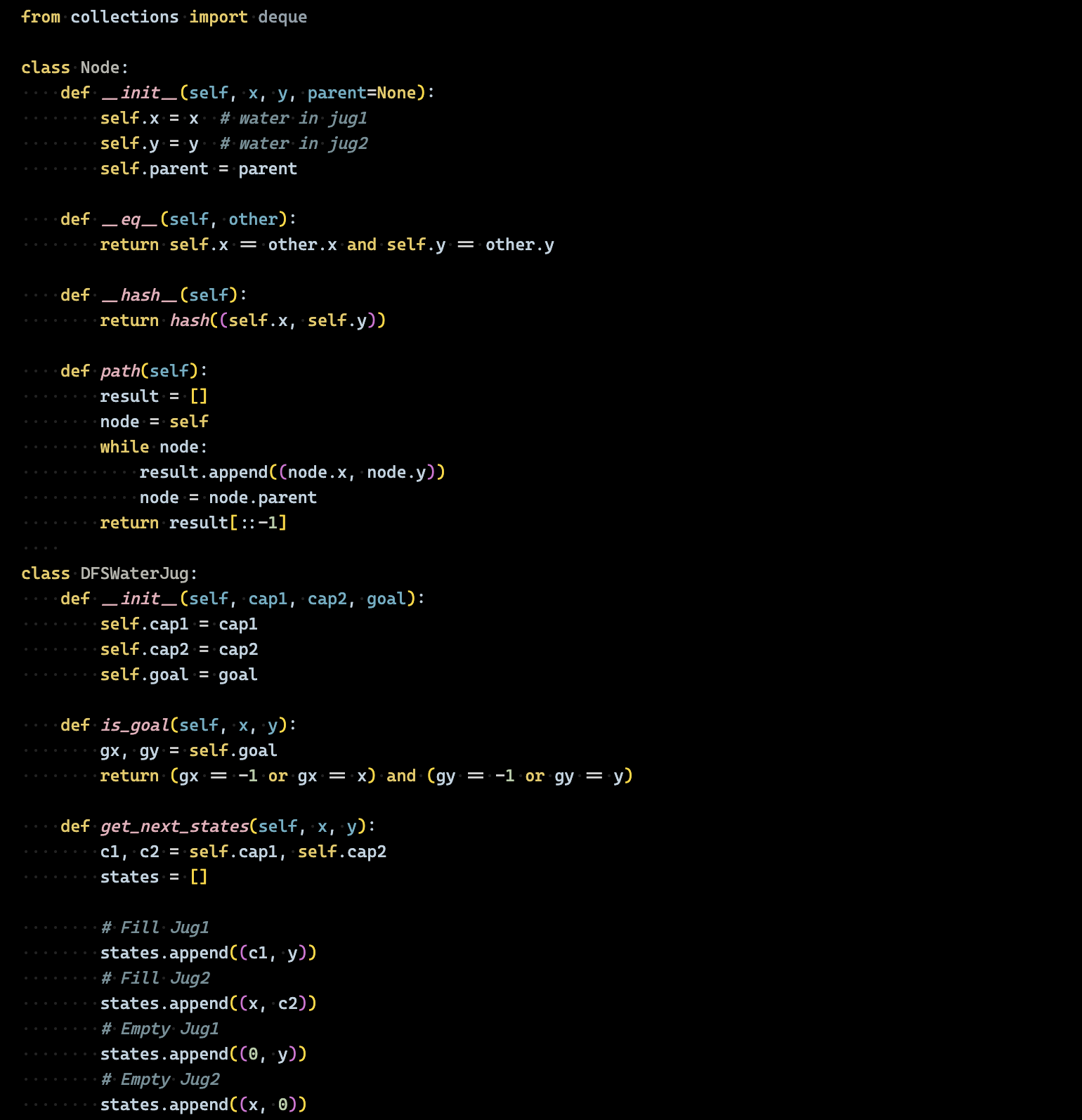
|  |
| --- |
|  |

|  |
| --- |
|  |
|  |

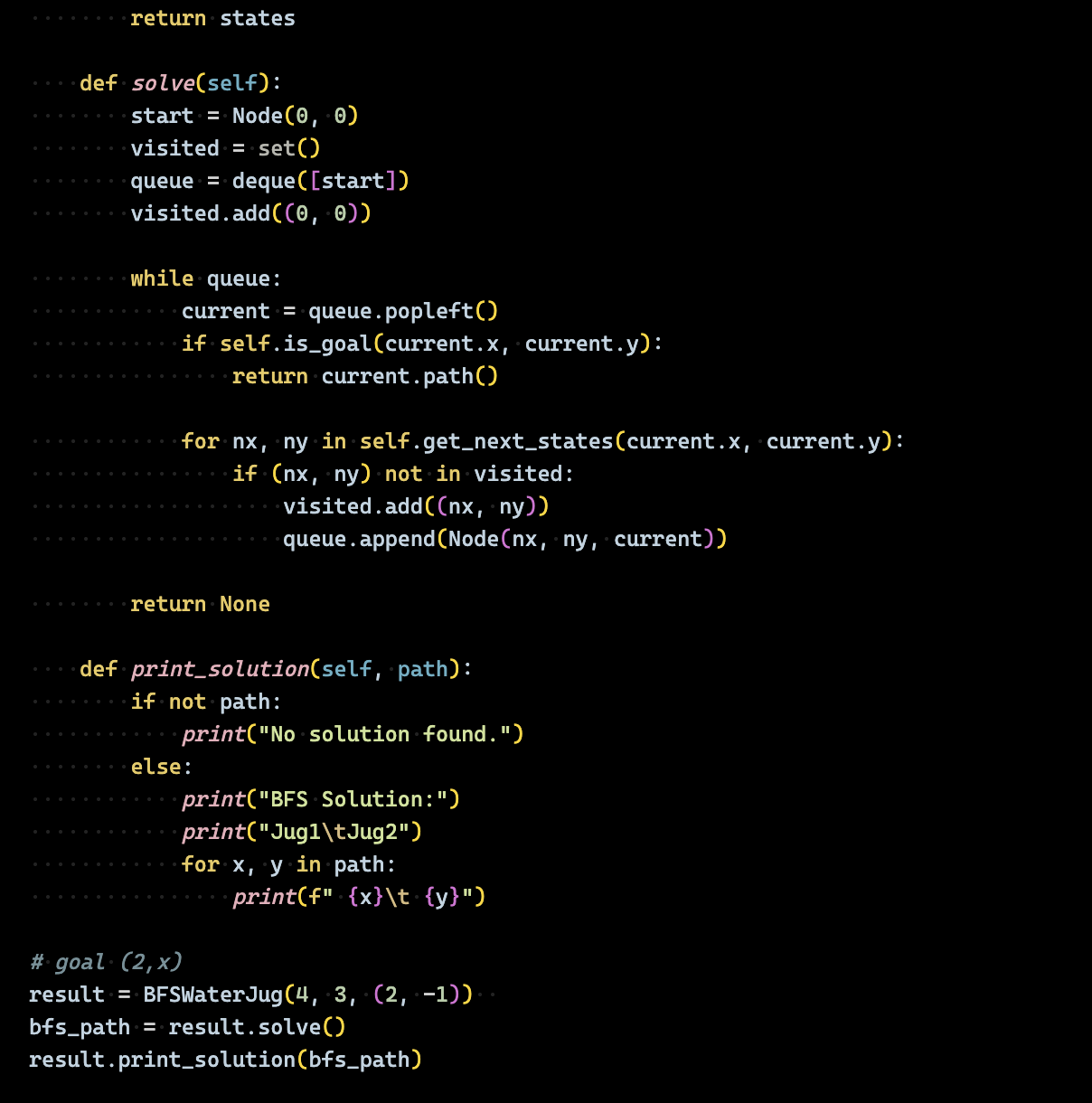
**Terminal Output:**

|  |
| --- |
|  |

**Code (Solution using DFS):**

**** A computer code with white text

AI-generated content may be incorrect.

****

**Terminal Output:**

|  |
| --- |
|  |

**Task-3:**

Comment on your experiment by a comparison between the results.

**Comparison Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Rule-Based** | **BFS** | **DFS** |
| Search Strategy | Rule-driven logic | Level-order traversal | Depth-order traversal |
| Guaranteed Shortest | No | Yes | No |
| Memory Usage | Low | High | Low |
| Goal Flexibility | Flexible | Flexible | Flexible |
| Output Clarity | Rule-based trace | Clean state sequence | Clean state sequence |

**Conclusion:**

The Water Jug Problem illustrates how different AI approaches yield different strengths:

* Rule-based systems are fast and structured but not always optimal.
* BFS guarantees the shortest solution but uses more memory.
* DFS is efficient in memory but may return longer or suboptimal paths.

For time-critical or explainable systems, rule-based logic is effective. For shortest path solutions, BFS is ideal.