|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **AY -ODD 2024-25** | | | | | | | | | | | | | |
| **GUJARAT TECHNOLOGICAL UNIVERSITY** | | | | | | | | | | | | | |
| **SCHOOL OF ENGINEERING AND TECHNOLOGY** | | | | | | | | | | | | | |
| **PRACTICAL - 4** | | | | | | | | | | | | | |
| **Course Code & Name** | | | **ME01095021- Artificial Intelligence** | | | | | | | | | | |
| **Academic Term:** | | | **AY –ODD 2024-25** | | | | | **Semester** | | | | **I** | |
| **Student Enrollment No:** | | | **241370795004** | | | | | **Batch:** | | | |  | |
| **Student Name:** | | | **Dake Darsh Dhaneshkumar** | | | | | | | | | | |
|  | |  |  |  | |  | |  | | | |  |  |
| **AIM/Objective:** | | | | | | | | | | | | | |
| 1 | | To implement A\* algorithm for Water-Jug Problem with local & global heuristic functions. | | | | | | | | | | | |
|  | |  |  |  | |  | |  | |  | | |  |
| **Expected Outcome:** | | | | | | | | | | **CO/PO/PSO** | | | |
| 1 | | Using the A\* algorithm for the Water Jug Problem with global and local heuristics, the algorithm efficiently finds the sequence of steps to measure the target amount of water while minimizing the evaluated cost. | | | | | | | | CO5 | | | |
|  | | **Experiment Result and Analysis**  **Resources and Software used:**   1. Python 3.12.6 2. PyCharm IDE   **Code:**  import heapq  # Function to perform A\* for the Water Jug Problem  def a\_star\_water\_jug(capacity1, capacity2, target, heuristic):  open\_list = []  parent = {} # To reconstruct the path  cost = {} # To store the cost to reach each state  # Initialize the starting state  start = (0, 0)  heapq.heappush(open\_list, (0, start))  parent[start] = None  cost[start] = 0  while open\_list:  \_, current\_state = heapq.heappop(open\_list)  jug1, jug2 = current\_state  # If the target is achieved  if jug1 == target or jug2 == target:  path = []  while current\_state is not None:  path.append(current\_state)  current\_state = parent[current\_state]  return path[::-1] # Reverse the path  # Generate possible moves  transitions = [  (capacity1, jug2), # Fill Jug 1  (jug1, capacity2), # Fill Jug 2  (0, jug2), # Empty Jug 1  (jug1, 0), # Empty Jug 2  (max(0, jug1 - (capacity2 - jug2)), min(capacity2, jug1 + jug2)), # Pour Jug 1 -> Jug 2  (min(capacity1, jug1 + jug2), max(0, jug2 - (capacity1 - jug1))) # Pour Jug 2 -> Jug 1  ]  for next\_state in transitions:  if next\_state not in cost or cost[current\_state] + 1 < cost[next\_state]:  cost[next\_state] = cost[current\_state] + 1  priority = cost[next\_state] + heuristic(next\_state, target)  heapq.heappush(open\_list, (priority, next\_state))  parent[next\_state] = current\_state  return None # Return None if no solution is found  # Heuristic functions  def global\_heuristic(state, target):  jug1, jug2 = state  return abs(target - jug1) + abs(target - jug2)  def local\_heuristic(state, target):  jug1, jug2 = state  return min(abs(target - jug1), abs(target - jug2))  # Example Usage  capacity1 = 3 # Capacity of the 3-liter jug  capacity2 = 4 # Capacity of the 5-liter jug  target = 2 # Target amount to measure  print("Using Global Heuristic:")  solution\_global = a\_star\_water\_jug(capacity1, capacity2, target, global\_heuristic)  if solution\_global:  print("Solution steps:", solution\_global)  else:  print("No solution found.")  print("\nUsing Local Heuristic:")  solution\_local = a\_star\_water\_jug(capacity1, capacity2, target, local\_heuristic)  if solution\_local:  print("Solution steps:", solution\_local)  else:  print("No solution found.")  **Output:** | | | | | | | |  | | |  |
|  | | |  |  | | --- | --- | | **Conclusion** | | | 1 | The A\* algorithm successfully solves the Water Jug Problem by leveraging heuristics to optimize the search. Global heuristics consider the total difference from the target, while local heuristics prioritize immediate progress. | | | | | | | | | | | | |
|  | |  |  |  |  | |  | |  | |  | | |
| **Evaluation Rubrics** | | | | | **Marks** | | **Inadequate** | | **Good** | | **Excellent** | | |
| **0%** | | **50%** | | **100%** | | |
| 1 | The understanding of the Student regarding the objective of the given practical | | | | **2** | |  | |  | |  | | |
| 2 | Installation of Software or Hardware Setup level | | | | **2** | |  | |  | |  | | |
| 3 | Quality of the Analysis done | | | | **2** | |  | |  | |  | | |
| 4 | Quality of the report including concluding remarks and Findings | | | | **2** | |  | |  | |  | | |
| 5 | Question & Answer related to given practical & timely submission | | | | **2** | |  | |  | |  | | |
|  | | | | | **10** | |  | |  | |  | | |
| **Total Marks Obtained Out of 10** | | | | | | |  | | | | | | |
|  | |  |  |  | |  |  | | | | |  |  |
|  | |  |  |  | |  |  | | | | |  |  |
|  | |  |  |  | |  |  | | | | |  |  |
|  | | **Date of Completion:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | |  | | **Course**  **Coordinator Sign:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | | | | | | | |