



Programming for Artificial Intelligence Lab

Submitted to:

Rasikh Ali

Submitted by:

Shazra Zainab 003

Department:

Software Engineering

Section:

BSAI(4A)

Roll No:

SU92-BSAIM-F23-003

Task 04

Input:

```

1  def water_puzzle(jugA_max, jugB_max, target, jugA=0, jugB=0, steps=0, visited=None):
2      if visited is None:
3          visited = set()
4      if (jugA, jugB) in visited:
5          return float('inf')
6      visited.add((jugA, jugB))
7      if jugA == target or jugB == target:
8          return steps
9      possible_moves = [
10         (jugA_max, jugB),
11         (jugA, jugB_max),
12         (0, jugB),
13         (jugA, 0),
14         (jugA - min(jugA, jugB_max - jugB), jugB + min(jugA, jugB_max - jugB)),
15         (jugA + min(jugB, jugA_max - jugA), jugB - min(jugB, jugA_max - jugA))
16     ]
17     return min(water_puzzle(jugA_max, jugB_max, target, a, b, steps + 1, visited) for a, b in possible_moves)
18
19 jug1_capacity = 4
20 jug2_capacity = 3
21 desired_amount = 2
22 result = water_puzzle(jug1_capacity, jug2_capacity, desired_amount)
23 print(f"Minimum steps required: {result}" if result != float('inf') else "No solution possible.")

```

Output:

```
def water_puzzle(jugA_max, jugB_max, target, jugA=0, jugB=0, steps=0, visited=None):  
    if target < 0 or target > max(jugA_max, jugB_max):  
        return None  
    if target == 0:  
        return steps  
    if visited is None:  
        visited = set()  
    state = (jugA, jugB)  
    if state in visited:  
        return None  
    visited.add(state)  
    # Fill A  
    if jugA_max > jugA:  
        new_state = (jugA_max, jugB)  
        steps += 1  
        result = water_puzzle(jugA_max, jugB_max, target, new_state[0], new_state[1], steps, visited)  
        if result is not None:  
            return result  
    # Fill B  
    if jugB_max > jugB:  
        new_state = (jugA, jugB_max)  
        steps += 1  
        result = water_puzzle(jugA_max, jugB_max, target, new_state[0], new_state[1], steps, visited)  
        if result is not None:  
            return result  
    # Empty A  
    if jugA > 0:  
        new_state = (0, jugB)  
        steps += 1  
        result = water_puzzle(jugA_max, jugB_max, target, new_state[0], new_state[1], steps, visited)  
        if result is not None:  
            return result  
    # Empty B  
    if jugB > 0:  
        new_state = (jugA, 0)  
        steps += 1  
        result = water_puzzle(jugA_max, jugB_max, target, new_state[0], new_state[1], steps, visited)  
        if result is not None:  
            return result  
    # Pour A to B  
    if jugA > 0 and jugB < jugB_max:  
        pour_amount = min(jugA, jugB_max - jugB)  
        new_state = (jugA - pour_amount, jugB + pour_amount)  
        steps += 1  
        result = water_puzzle(jugA_max, jugB_max, target, new_state[0], new_state[1], steps, visited)  
        if result is not None:  
            return result  
    # Pour B to A  
    if jugB > 0 and jugA < jugA_max:  
        pour_amount = min(jugB, jugA_max - jugA)  
        new_state = (jugA + pour_amount, jugB - pour_amount)  
        steps += 1  
        result = water_puzzle(jugA_max, jugB_max, target, new_state[0], new_state[1], steps, visited)  
        if result is not None:  
            return result  
    return None
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [] [X] ... ^ X

- PS C:\Users\DELL\Desktop\PAI Task 03> & C:/Users/DELL/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/DELL/Desktop/P
AI Task 03/water_jug.py"
Minimum steps required: 6
- PS C:\Users\DELL\Desktop\PAI Task 03>

