|  |  |
| --- | --- |
| **EX.NO : 02** | **WATER JUG PROGRAM USING BFS** |
| **DATE : 06.03.2024** |

**PROBLEM STATEMENT:**

You are given the 2 jugs, A 4-gallon jug and 3-gallon jug neither has any measuring markers on it. There is a pump that can be used to fill the jug with water how can you get two gallons of water into the 4-gallon jug.

**AIM:**

To solve the Water jug Problem using Breadth First Search.

**ALGORITHM:**

Step 1: Start.

Step 2: Get the capacity of A jug and B jug and Target.

Step 3: Create water jug problem function.

Step 4: In Function A, B, Target are parameter.

Step 5: In Function initialize state = (0, 0).

Step 6: Create parent empty set.

Step 7: Create frontier isn’t there is all the possible states to will be stored.

Step 8: Using while loop frontier is the condition.

Step 9: Assign state = frontier pop.

Step 10: If state is reaching the target.

**PROGRAM:**

from collections import deque def BFS(a, b, target):

m = {}

isSolvable = False path = []

q = deque() q.append((0, 0))

while len(q) > 0:

u = q.popleft() # Use popleft to get the first element (breadth-first) if (u[0], u[1]) in m:

continue

if u[0] > a or u[1] > b or u[0] < 0 or u[1] < 0: continue

path.append([u[0], u[1]])

m[(u[0], u[1])] = 1

if u[0] == target or u[1] == target: isSolvable = True

if u[0] == target: if u[1] != 0:

path.append([u[0], 0]) else:

if u[0] != 0: path.append([0, u[1]])

sz = len(path)

for I in range(sz):

print(“(“, path[i][0], “,”, path[i][1], “)”)

return # Exiting the function after finding the solution q.append([u[0], b])

q.append([a, u[1]])

for ap in range(max(a, b) + 1):

c = u[0] + ap

d = u[1] – ap

if c == a or (d == 0 and d >= 0): q.append([c, d])

c = u[0] - ap

d = u[1] + ap

if (c == 0 and c >= 0) or d == b:

q.append([c, d])

q.append([a, 0])

q.append([0, b])

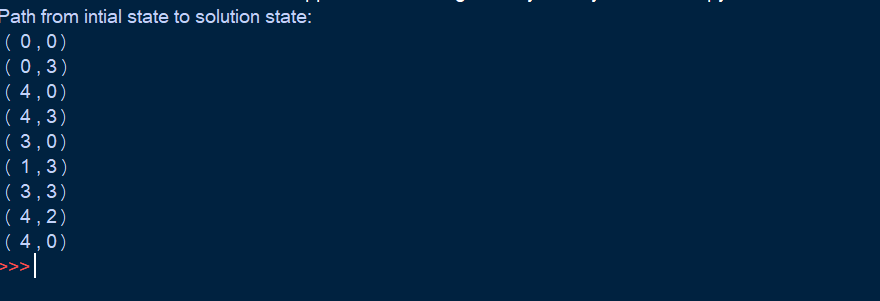
if not isSolvable: print("No solution")

if name == ' main ':

Jug1, Jug2, target = 4, 3, 2

print("Path from initial state to solution state:") BFS(Jug1, Jug2, target)

**OUTPUT:**



**RESULT:**

Thus, to solve the Water jug Problem using Breadth First Search has been executed successfully.