

```

import matplotlib.pyplot as plt
import networkx as nx
from collections import deque

class Estado:
    def __init__(self, jarra_4_litros, jarra_3_litros):
        self.jarra_4_litros = jarra_4_litros
        self.jarra_3_litros = jarra_3_litros

    def __eq__(self, other):
        return (self.jarra_4_litros, self.jarra_3_litros) == (other.jarra_4_litros, other.jarra_3_litros)

    def __hash__(self):
        return hash((self.jarra_4_litros, self.jarra_3_litros))

    def __str__(self):
        return "{}, {}".format(self.jarra_4_litros, self.jarra_3_litros)

    def es_objetivo(self):
        return self.jarra_4_litros == 2

    def transiciones(self):
        # Llenar la jarra de 4 litros.
        yield Estado(4, self.jarra_3_litros)
        # Llenar la jarra de 3 litros.
        yield Estado(self.jarra_4_litros, 3)
        # Vaciar la jarra de 4 litros.
        yield Estado(0, self.jarra_3_litros)
        # Vaciar la jarra de 3 litros.
        yield Estado(self.jarra_4_litros, 0)
        # Verter de la jarra de 3 litros a la de 4 litros.
        total = self.jarra_3_litros + self.jarra_4_litros
        yield Estado(total if total <= 4 else 4, 0 if total <= 4 else total - 4)
        # Verter de la jarra de 4 litros a la de 3 litros.
        total = self.jarra_3_litros + self.jarra_4_litros
        yield Estado(0 if total <= 3 else total - 3, total if total <= 3 else 3)

def bfs(inicio):
    G = nx.DiGraph() # Crear el grafo
    visitados = set()
    cola = deque([(inicio, [])])

    while cola:
        estado, camino = cola.popleft()

        if estado.es_objetivo():
            return G, camino + [estado]

        visitados.add(estado)

        for nuevo_estado in estado.transiciones():
            if nuevo_estado not in visitados:
                cola.append((nuevo_estado, camino + [estado]))
                G.add_edge(str(estado), str(nuevo_estado)) # Agregar el nuevo estado al grafo

    return None # No se encontró una solución.

def dibujar_grafo(G):
    plt.figure(figsize=(8,6))
    pos = nx.shell_layout(G) # positions for all nodes
    nx.draw(G, pos, with_labels=True, node_color='lightblue', node_size=1500, font_size=16, edge_color='gray')

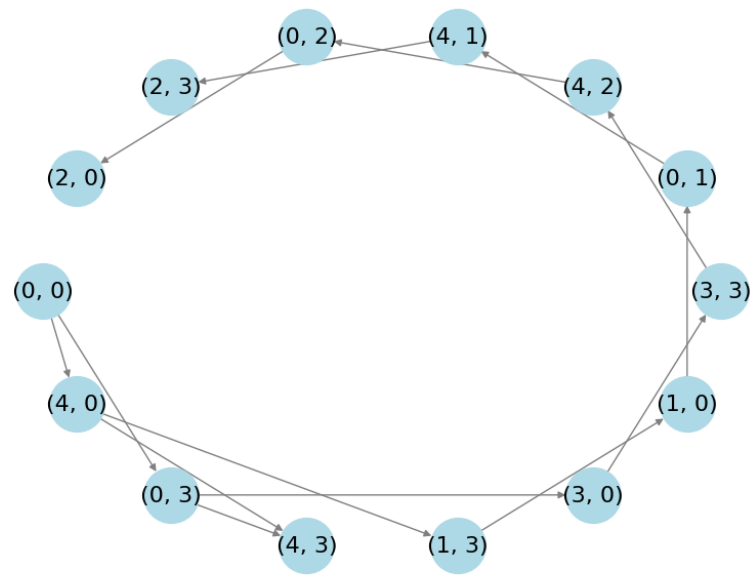
inicio = Estado(0, 0)
G, camino = bfs(inicio)

if camino is None:
    print('No se encontró una solución.')
else:
    print('Solución:')
    for estado in camino:
        print(estado)
    dibujar_grafo(G) # Dibujar el grafo
    plt.show() # Mostrar el grafo

```



```
Solución:  
(0, 0)  
(4, 0)  
(1, 3)  
(1, 0)  
(0, 1)  
(4, 1)  
(2, 3)
```



Haz doble clic (o pulsa Intro) para editar

Haz doble clic (o pulsa Intro) para editar