

_4bucles/_3laberinto.py

```
1 # Ejercicio 3: Problema de Resolución de Laberinto:
2
3 # Imagina que eres parte de un equipo de desarrollo de IA que se encarga de crear un sistema
  para que un robot resuelva laberintos. El laberinto está representado por una matriz, donde
  ciertos valores indican caminos permitidos (0), paredes (1), y la salida (9). Tu tarea es
  implementar una función recursiva que encuentre la ruta más corta para que el robot salga del
  laberinto.
4
5 # Toma en cuenta los siguientes puntos:
6
7 # La matriz representa el laberinto, donde los valores son:
8 # 0: Camino permitido.
9 # 1: Pared, no se puede atravesar.
10 # 9: Salida del laberinto.
11 # Debes implementar la función resolver_laberinto que utiliza recursividad para encontrar la
  ruta más corta desde una posición inicial hasta la salida.
12 # La función debe devolver una lista de coordenadas que representan la ruta desde la posición
  inicial hasta la salida.
13 # Puedes usar una lista de movimientos posibles: arriba ((-1, 0)), abajo((1, 0)), izquierda
  ((0, -1)), derecha ((0, 1)).
14
15 def resolver_laberinto(laberinto, fila, columna, camino=None):
16     if camino is None:
17         camino = []
18
19     if not(0<=fila<len(laberinto)) or not (0<=columna<len(laberinto[0])) or laberinto[fila]
[columna]==1 or (fila,columna) in camino:
20         return None
21     camino.append((fila,columna))
22
23     #caso base ///// si ya nos encontramos ne la salida
24
25     if laberinto[fila][columna]==9:
26         return camino
27
28     #definir los movimientos posibles
29     movimientos=[(-1,0),(1,0),(0,-1),(0,1)]
30
31     for movimiento in movimientos:
32         nueva_fila,nueva_columna=fila+movimiento[0],columna+movimiento[1]
33         resultado = resolver_laberinto(laberinto,nueva_fila,nueva_columna,camino.copy())
34         if resultado:
35             return resultado
36     return None # No hay una solucion desde esta posicion.
37
38 def imprimir_laberinto(laberinto, camino):
39     for fila in range(len(laberinto)):
40         for columna in range(len(laberinto[0])):
41             if (fila,columna) in camino:
42                 print("*",end=" ") #Representa el camino
43
44             else:
```

```
45     print(laberinto[fila][columna],end=" ")
46     print()
47
48     laberinto = [
49         [0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
50         [1, 1, 0, 1, 0, 1, 1, 1, 1, 0],
51         [0, 1, 0, 1, 0, 0, 0, 0, 0, 0],
52         [0, 1, 0, 0, 0, 1, 1, 1, 1, 0],
53         [0, 1, 1, 1, 1, 1, 0, 0, 1, 0],
54         [0, 0, 0, 0, 0, 0, 0, 1, 1, 0],
55         [0, 1, 1, 1, 1, 1, 1, 1, 0, 0],
56         [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
57         [1, 1, 1, 1, 0, 1, 1, 1, 9, 1],
58         [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
59     ]
60     camino_solucion = resolver_laberinto(laberinto,0,0)
61
62     if camino_solucion:
63         print("El camino para salir del laberinto es:")
64         imprimir_laberinto(laberinto,camino_solucion)
65     else:
66         print('No hay solucion para este laberinto')
67
```