PROBLEMA DE MAZE

El problema de Maze, establece el problema de un robot que busca salir de un laberinto con una representación en 2D de la siguiente forma:

```
maze = [
    ["#", "#", "#", "#", "#", "#", "#","#"],
    ["#", "S", "#", " ", "#", " ", "E","#"],
    ["#", " ", " ", " ", "#", " ", " ","#"],
    ["#", " ", "#", " ", " ", " ", "#","#"],
    ["#", "#", "#", "#", "#", "#", "#","#"],
    ["#", "#", "#", "#", "#", "#", "#","#"]
]
```

Donde S representa la posición inicial del robot, y E la salida,
representa paredes que el robot no puede atravesar,

" ", representan espacios en blanco que el robot puede usar para moverse.

El algoritmo actual hace uso de una heurística llamada Distancia manhattan que utiliza las coordenadas actuales del robot y las coordenadas de la salida, con el fin de establece un valor numérico correspondiente a la distancia usando líneas horizontales y verticales entre ambos puntos.

Otra posibilidad, puede ser hacer uso de la distancia euclidiana, que establece la distancia en línea recta, desde el punto actual del robot hasta la salida.

El docente le ha suministrado un código con varias pistas, el cual usted debe intervenir y dejar funcional.

Su misión es:

- Completar el código de Maze_problem.ipynb.

- Identificar claramente la estructura del problema y cada una de las funciones: actions, result, action-cost.
- Completar el código para encontrar la solución al laberinto.
- ¿Cómo cambia el comportamiento del algoritmo si cambiamos la función de costo?
- ¿Qué sucede si hay múltiples salidas en el laberinto? ¿Cómo podrías modificar el algoritmo para manejar esto? o Modifica el laberinto por uno más grande y con otros tipos de obstáculos además de paredes.
- ¿Qué limitaciones encuentras en el algoritmo? Nota: Resuelve este problema en una celda aparte para mantener la integridad de tu código original.

Notas Adicionales:

- Teniendo en cuenta que el robot se esta moviendo en un espacio de 2D, las acciones disponibles deberían considerar moverse hacia arriba, abajo, derecha e izquierda, no tenga en cuenta movimientos diagonales.
- Tenga en cuanta que cada movimiento realizado por el robot debe cambiar su posición, las posiciones son dadas como una coordenada de x y de y de la siguiente forma (x,y)
- La posición del robot corresponde al state
- Todas las acciones (Up, Down, Right, Left) tienen el mismo costo (1), por lo que la mejor acci+on posible debería estar guiada por la heurística (Manhattan o Euclidean distance).

- Realmente, la elección de la heurística depende de las restricciones de movimiento del agente (solo ortogonales vs. también diagonales).
- Este problema también podría resolverse por una estrategia ciega Breadth First Search, pero con mayor costo temporal y espacial