

● Complete los siguientes datos

- Número de grupo: 02
- Nombre y Apellidos de los integrantes: Jon Ochoa, Xabier Lopez
- Entrega: 4
- Fecha: 20/11/2025
- Tiempo dedicado a la entrega: 25 Horas aprox

● Notas generales

- Dos o más prácticas iguales o con un índice de similitud muy alto serán consideradas plagio por todos los implicados.
- Todas las prácticas son susceptibles de ser revisadas minuciosamente, incluyendo una defensa por parte de sus autores de forma individual. En caso de suspender esta defensa se suspendería la práctica.
- La práctica supone la realización de todos los apartados y el cumplimiento de todas las condiciones, si falta alguno, la práctica estará incompleta y no será evaluada.
 - **Las prácticas que no cumplan este requisito serán consideradas no válidas y obtendrán una calificación de 0.**
- Las prácticas han de ser funcionales y no dar error en su ejecución.
 - **Las prácticas que no cumplan este requisito serán consideradas no válidas y obtendrán una calificación de 0.**
- El uso de código sin referenciar implicará la consideración de la práctica como no apta.

● Instrucciones

Este es el cuaderno que utilizarás como **plantilla** para entrega de la práctica correspondiente.

Por favor, **lee atentamente** el enunciado. Si tienes alguna duda, utiliza el foro o pregunta en clase (pero no compartas código).

Consideraciones generales sobre el código:

- Utilizar tantas celdas (de texto o de código) como considere para dar una respuesta legible y clara a las preguntas planteadas.
- Mostrar resultados intermedios o finales (tablas, contenido de las variables...) que demuestren que la solución es correcta si fuera necesario.

- No olvidéis importar todas las librerías necesarias para la correcta ejecución del código.

En las indicaciones, hay instrucciones extra indicadas con el símbolo . No son requisitos para superar la práctica, sino acciones que puedes realizar para mejorar la calificación.

La calificación de los ejercicios se hará con los siguientes criterios:

- [9 a 10] puntos: La solución aportada cumple con los requisitos de forma excelente y existen aspectos positivos o extras () que avalan su excelencia.
- [7 a 9] puntos: La solución aportada cumple con los requisitos, pero algunas cuestiones menores son susceptibles de ser mejoradas, tales como: presentación de la solución, justificación de la decisión tomada, eficiencia del código...
- [5 a 7] puntos: Las soluciones aportadas no cumplen alguno de los requisitos, como por ejemplo: no se responde a una pregunta, no se aporta la solución a una parte...
- [1 a 5] puntos: La solución aportada no cumple con varios requisitos.
- [0] puntos: solución no aportada o solución plagiada. **Cabecera no completada, ficheros con nombre erróneo, etc.**

La nota de la práctica dependerá de:

1. Cumplimiento de las condiciones.
2. Sofisticación del sistema.
3. Limpieza y eficiencia del código.
4. Comentarios aportados.



Desarrollo a realizar

En esta práctica tendrás que desarrollar un sistema inteligente relacionado con la [temática seleccionado](#) al inicio del curso.

El problema a resolver ha de ser solucionable a partir de problema de búsquedas.

Define un escenario que requiera algoritmos de búsqueda (por ejemplo, rutas de entrega entre varios puntos, localización de recursos, etc.). El proyecto debe incluir más de una búsqueda, no solo un único recorrido o punto de destino.

La elección de la temática y su correcta representación en Python será clave para poder resolver la práctica con éxito.



Entrega

Validación del problema

Descripción del problema a resolver:

El objetivo de esta práctica es desarrollar un sistema de búsqueda en un entorno tridimensional que permita a un robot móvil (AMR) desplazarse dentro de un almacén automatizado desde una posición inicial hasta un destino final, evitando obstáculos y utilizando un ascensor o rampa para cambiar de nivel.

El robot parte desde la zona de carga (S) en el nivel 0 y debe llegar al punto de entrega (O) en el nivel 1, planificando su ruta más corta en términos de coste total. El movimiento se realiza en las cuatro direcciones cardinales (arriba, abajo, izquierda, derecha) dentro de cada nivel, y el cambio de nivel solo es posible a través del punto A (ascensor).

El problema se formula como un problema de búsqueda en grafos, donde cada celda libre del mapa representa un nodo del grafo y los movimientos posibles definen las aristas entre estados. El coste de cada desplazamiento es unitario, excepto el cambio de nivel en el ascensor, que tiene un coste adicional de 5 unidades.

Objetivo: encontrar la secuencia de movimientos con menor coste total (camino óptimo) desde el punto inicial S hasta el punto objetivo O, considerando los obstáculos, los dos niveles y el coste extra de transición entre plantas.

- Confirmación del docente (fecha y comentarios):

Confirmado a dia 10/11/2025

Representación de estados propuesta:

El mapa tendrá 2 pisos, en los que los "." representarán un camino transitado y los "#" obstáculos. Las "A" representarán ascensores por los que se podrá acceder a los demás pisos.

```
In [100]: planta0 = [
    ['S', '.', '.', '.', '.', '.', '.', '.', '.', '.', '.', '.'],
    ['.', '#', '#', '#', '.', '.', '.', '#', '#', '#', '.'],
    ['.', '.', '.', '#', '.', '.', '.', '#', '.', '.', '.'],
    ['.', '#', '.', '#', '#', '#', '.', '#', '.', '.', '.'],
    ['.', '#', '.', '.', '.', '#', '.', '#', '.', '.', '.'],
    ['.', '#', '.', '.', '#', '#', '.', '#', '.', '.', '.'],
    ['.', '#', '#', '#', '.', '#', '.', '.', '.', '.', '.'],
    ['.', '#', '#', '#', '.', '.', '.', '#', '#', '#', '.'],
    ['.', '#', '#', '#', '.', '#', '#', '#', '#', '#', 'A'],
    ['.', '.', '#', '#', '.', '#', '#', '#', '#', '#', '.'],
    ['.', '.', '#', '#', '.', '.', '.', '#', '#', '#', '.']
]

planta1 = [
    ['.', '.', '.', '.', '.', '.', '.', '.', '.', '.', '.'],
    ['.', '#', '#', '#', '.', '#', '.', '.', '.', '.'],
    ['.', '.', '.', '#', '.', '#', '.', '#', '.', '.'],
    ['.', '#', '.', '#', '#', '#', '.', '#', '.', '.'],
    ['.', '#', '.', '.', '.', '#', '.', '#', '.', '.'],
    ['.', '.', '#', '#', '#', '#', '.', '#', '.', '.'],
    ['.', '#', '.', '.', '.', '#', '#', '#', '#', '.'],
    ['.', '.', '.', '#', '#', '#', '#', '#', '#', 'A'],
    ['.', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
    ['.', '.', '.', '#', '#', '#', '#', '#', '#', '#', 'O', '#', '.']
]
```

]

```
plantas = [planta0, planta1]
```

- Confirmación del docente (fecha y comentarios):

Confirmado a dia 10/11/2025, Comentarios: añadir iconos para representar algunos de los nodos, que no sean todo puntos

Version 2 del mapa: Confirmado a dia 12/11/2025

Símbolo | Significado

----- -----
--
E Elevador que conecta los niveles A, B, X Estantrias / Objetivos en la planta 0
D...L Estanterías / Objetivos en la planta 1
C, R, U Estaciones de carga / Casillas de salida en la planta 0 (Casillas de salida seleccionables)
Obstáculo fijo (muro, maquinaria, zona no accesible)
T Zona lenta (tráfico alto, suelo irregular o pasillo estrecho; coste 2)
. Zona libre / pasillo transitable (coste 1)

In [101...]

#Version 2 del mapa:

```
# planta0 = [
# ['C', '.', '.', '.', '.', '#', '#', '.', '.', '.', '.', 'E'],
# ['.', 'A', '#', '.', 'T', '.', '.', '#', '.', '.', '.'],
# ['.', '.', '#', '.', '.', '.', 'B', '.', '.', 'T', '.'],
# ['.', 'E', '#', '.', '#', '.', '.', '.', '.', '.', '.'],
# ['.', '.', '.', '.', '.', '.', '#', 'U', '.', '.'],
# ['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.'],
# ['.', '.', 'T', '.', '#', '.', '.', '.', '.', '.'],
# ['.', '#', '.', 'R', '.', '.', '#', '.', '.', '.'],
# ['.', '.', '.', '#', '.', '#', '.', '.', '.', '.'],
# ['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
# ]

# planta1 = [
# ['.', '.', '.', '.', '.', '.', 'J', '.', '.', '.', 'E'],
# ['.', 'D', '.', 'T', '.', '.', '.', '#', '.', '.', '.'],
# ['.', '.', '#', '.', '.', '.', '.', '#', '.', '.', '.'],
# ['.', 'E', '#', '.', '#', '.', '.', '.', '.', '.'],
# ['.', '.', '.', 'F', '.', '.', '.', '#', '.', '.', '.'],
# ['.', '.', '#', '.', '.', 'E', '.', '.', '#', '.'],
# ['.', '.', 'T', '.', '#', '.', 'G', '.', 'H', '.'],
# ['.', '#', '.', '.', '.', '#', '.', 'I', '.'],
# ['.', '.', '.', '#', '.', '#', '.', '.', '.', '.'],
# ['.', '.', '.', 'T', '.', '.', 'K', '.', 'L', '.']
# ]
```

#Usar la anterior para que el print del mapa salga bien

```

planta0 = [
    ['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E'],
    ['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.'],
    ['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.'],
    ['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.'],
    ['.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.'],
    ['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.'],
    ['.', '.', 'T', '.', '#', '.', '.', '.', '.', '.'],
    ['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.'],
    ['.', '.', '.', '#', '.', '#', '.', '.', '.', '.'],
    ['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
]

planta1 = [
    ['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E'],
    ['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.'],
    ['.', '.', '#', '.', '.', '.', '.', '#', '.', '.'],
    ['.', 'E', '#', '#', '.', '#', '.', '.', '.'],
    ['.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.'],
    ['.', '.', '#', '.', '.', 'E', '.', '.', '#', '.'],
    ['.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.'],
    ['.', '#', '.', '.', '.', '#', '.', '.', '#', 'Estanteria I', '.'],
    ['.', '.', '.', '#', '.', '#', '.', '.', '.', '.'],
    ['.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
]

#Supuestamente escalable a todos los pisos que se quieran
plantas = [planta0, planta1]

```

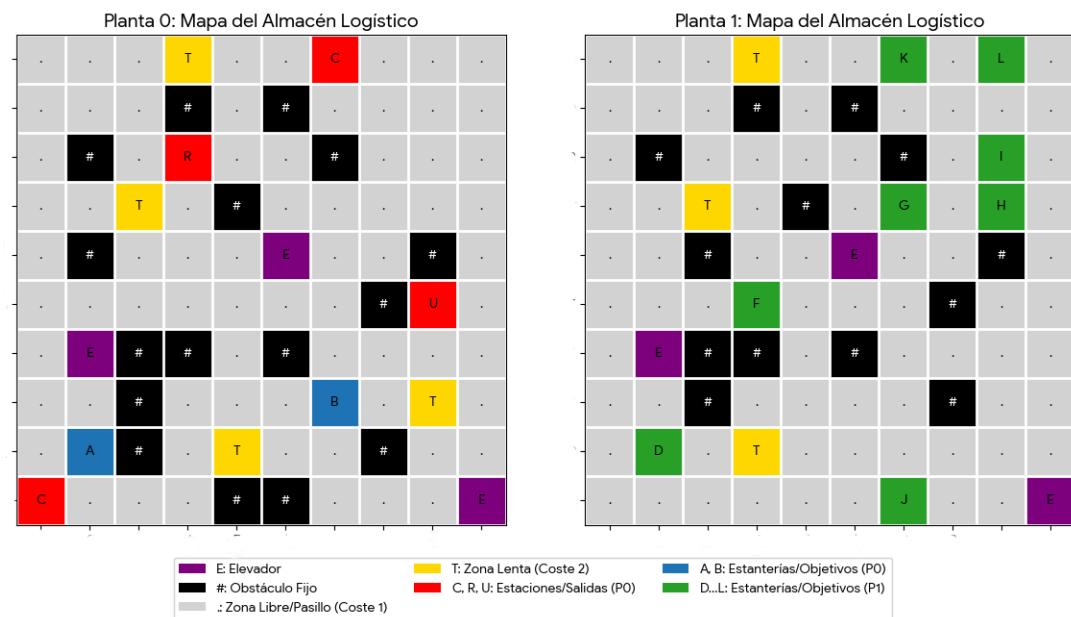
Introducción

Breve descripción del problema.

El trabajo desarrolla un sistema de búsqueda para un robot móvil (AMR) que navega dentro de un almacén automatizado compuesto por dos niveles, cada uno modelado mediante una matriz de 10×10 . El objetivo del robot es viajar desde una estación de salida seleccionable (como C, R, o U) por una serie de estanterías objetivo (por ejemplo F, G, L...), evitando obstáculos, atravesando zonas lentas y utilizando ascensores (E) para cambiar de planta. El movimiento permitido es únicamente en las cuatro direcciones cardinales, siguiendo un orden explícito URDL (Up, Right, Down, Left). Cada movimiento tiene un coste asociado dependiendo del tipo de terreno, y el uso del ascensor añade un coste fijo adicional.

En este contexto, el problema se formaliza como un problema de búsqueda en grafos, en el que cada celda transitable representa un estado y cada movimiento válido una transición. El objetivo es encontrar la ruta más barata(supuestamente en tiempo) desde el origen hasta el destino utilizando algoritmos clásicos de búsqueda.

Mapa del almacén propuesto [Generado con Gemini]:



Contexto y finalidad del sistemas inteligente a implementar.

El sistema simula el comportamiento de un robot real dentro de un almacén logístico, donde es habitual trabajar con varios pisos, ascensores, obstáculos, zonas de acceso restringido y diferentes áreas de recogida. La planificación óptima de rutas no solo reduce tiempos, sino también consumo energético y congestión.

Este sistema permite:

- Comparar algoritmos informados (Greedy, A*) y no informados (BFS, DFS).
- Analizar cómo influye el orden explícito de exploración en los algoritmos sin heurística.
- Observar el impacto de las heurísticas multinivel, especialmente cuando existen varios ascensores.
- Realizar búsquedas variadas seleccionando distintos puntos de partida y estanterías objetivo.

Programa

Requisitos

- Utilizar la clase Nodo explicada en clase con los métodos y atributos vistos en los ejercicios prácticos.
- Emplear métodos getters y setters como los requeridos en los ejercicios propuestos.
- En la representación del problema han de existir obstáculos.
- En la representación del problema han de existir varios niveles.
- Utiliza una matriz (sin librerías) de al menos 2 niveles de 10x10.
- En la representación del problema deben existir varios recursos u objetivos que actúen como base para realizar múltiples búsquedas (por ejemplo, varios paquetes/envíos, ubicaciones a visitar, elementos a recoger, etc.).
- Resolver el problema empleando las versiones Graph de BFS, DFS, Greedy y A*.

8. Define **explícitamente** un orden de visita en el caso de los algoritmos no informados.
9. Por cada iteración del algoritmo muestra **los nodos expandidos** y la frontera.
10. Finalmente muestra la ruta encontrada y el coste.
11. Se han de relizar al menos 3 búsquedas.
12. **No es posible utilizar librerías.**

Consideraciones deseables

- Desarrolla las versiones Tree de los algoritmos.
- Utilizar dos versiones de A* y Greedy con 2 heurísticas.
- El sistema ha de tener un grado de sofisticación elevado.

Requisitos extra

- Proporcionar una representación visual de la ruta encontrada y de los nodos expandidos durante la búsqueda.
- Comparar el rendimiento de los algoritmos (tiempo, nodos expandidos, coste de la ruta).

1) Código

Clase nodo

```
In [102...]: class Nodo:
    def __init__(self, estado=None, padre=None, accion=None, acciones=None, coste_acumulado=0, fn=0, heuristic=0, coordenadas = None):
        self._estado = estado
        self._padre = padre
        self._accion = accion or []
        self._acciones_disponibles = acciones or []
        self._coste = coste
        self._coste_acumulado = coste_acumulado
        self._heuristic = heuristic
        self._fn = fn
        self._profundidad = profundidad
        self._coordenadas = coordenadas

    def get_estado(self): return self._estado
    def set_estado(self, e): self._estado = e

    def get_padre(self): return self._padre
    def set_padre(self, p): self._padre = p

    def get_accion(self): return self._accion
    def set_accion(self, lista): self._accion = list(lista)

    def get_coste_acumulado(self): return self._coste_acumulado
    def set_coste_acumulado(self, coste_acumulado): self._coste_acumulado = coste_acumulado

    def get_heuristic(self): return self._heuristic
    def set_heuristic(self, heuristic): self._heuristic = heuristic

    def get_fn(self): return self._fn
    def set_fn(self, fn): self._fn = fn
```

```

def get_acciones_disponibles(self): return self._acciones_disponibles
def set_acciones_disponibles(self, lista): self._acciones_disponibles = lista

def get_coste(self): return self._coste
def set_coste(self, c): self._coste = c

def get_profundidad(self): return self._profundidad
def set_profundidad(self, p): self._profundidad = p

def get_coordenadas(self): return self._coordenadas
def set_coordenadas(self,c): self._coordenadas = c

def __str__(self):
    padre_estado = self._padre.get_estado() if self._padre else None

    return (
        f"Nodo {self._estado}\n"
        f" Estado: {self._estado}\n"
        f" Coordenadas: {self._coordenadas}\n"
        f" Padre: {padre_estado}\n"
        f" Acción: {self._accion}\n"
        f" Acciones disponibles: {self._acciones_disponibles},\n"
        f" Coste: {self._coste}\n"
        f" Coste Acumulado: {self._coste_acumulado}\n"
        f" Heuristica: {self._heuristica}\n"
        f" F(n): {self._fn}\n"
        f" Profundidad: {self._profundidad}\n"
        f")"
    )

```

```

In [103...]: # --- Parámetros del entorno ---

COST_ASC = 5 # coste de usar elevador
COST_LENT = 2 # coste zonas lentes "T"
COST_NORM = 1 # coste zonas normales "." / Estanterias
PROFUNDIDAD_MAX = 1000 #Profundidad maxima permitida antes de que lo marque como
FRONTERA_MAX = 1500
LIMITE_PRINT_FRONTERA = 50 #Limite de registros que se muestran de La frontera,

debug = False # activar / desactivar debug
mostrar_iteracion = True #Activar/ Desactivar el imprimir la frontera y el nodo

PLANTAS = len(plantas)
FILAS = len(plantas[0])
COLUMNAS = len(plantas[0][0])

ORIGEN = ()
DESTINO = ()
ASCENSORES = []
MARCAS = []

```

Funciones de ayuda

```

In [104...]: def fn_a(nodo):
    #Modificar esto segun el tipo de busqueda

```

```

    return nodo.get_heuristica() + nodo.get_coste_acumulado()

def fn_greedy(nodo):
    return nodo.get_heuristica()

def clave_orden(nodo):
    return nodo.get_fn()

def imprimir_frontera(frontera):
    # Limitar la impresión a los primeros 200 nodos
    LIMITE_PRINT_FRONTERA
    frontera_límite = frontera[:LIMITE_PRINT_FRONTERA]

    for i in frontera_límite:
        print(f"{{i.get_estado()}}, {i.get_fn()})", end=", ")

    # Si la frontera es más grande que el límite, imprime un mensaje indicando q
    if len(frontera) > LIMITE_PRINT_FRONTERA:
        print(f"...(se han mostrado solo los primeros {LIMITE_PRINT_FRONTERA} no

#Funcion que pasadas las coordenadas devuelve lo que representa la casilla y su c
def tipo_casilla(plan, x, y):

    casilla = plantas[plan][x][y]
    coste = 0

    if casilla == "T":
        coste = COST_LENT
    else:
        coste = COST_NORM

    return casilla, coste

```

Funciones de mapeado:

In [105...]

```

#Mapeado de las posiciones clave para poder calcular la heuristica
def meapeado_inicial(origen,destino):
    #Usamos las variables globales por comodidad
    global ORIGEN, DESTINO, ASCENSORES,MARCAS

    for planta in range(PLANTAS):
        for x in range(FILAS):
            for y in range(COLUMNAS):

                if plantas[planta][x][y] == origen:
                    ORIGEN = (planta,x,y)

                elif plantas[planta][x][y] == destino:
                    DESTINO = (planta,x,y)

                elif plantas[planta][x][y] == "E":
                    if (x,y) not in ASCENSORES:
                        ASCENSORES.append((x,y))
                    if "E" not in MARCAS:
                        MARCAS.append("E")

                elif plantas[planta][x][y] != ".":


```

```

    if (plantas[planta][x][y]) not in MARCAS:
        MARCAS.append(plantas[planta][x][y])
if debug:
    print(f"Punto de los ascensores: {ASCENSORES}")
    print(f"Punto de origen {ORIGEN}")
    print(f"Punto de destino {DESTINO}")

```

Funciones de solucion

In [106...]

```

#Solucion con todos Los parametros
def solucion(nodo,visitados):

    #Si no hay solucion
    if nodo == None:
        print("No hay ruta posible entre esos puntos")
        return None

    print("----- RESULTADO DE LA BUSQUEDA -----")
    print(f"El nodo resultante es:")
    print(nodo)
    acciones = nodo.get_accion()
    print(f"El camino recorrido es: {acciones}")
    print(f"El coste total es de {nodo.get_coste_acumulado()}")
    print(f"La cantidad de nodos expandidos es de {len(visitados)}")

    #Sacar el primero y el ultimo para poder ver el inicio y el final

    copia_mapa = []

    for planta in plantas:
        copia_planta = []
        for fila in planta:
            copia_fila = []
            for celda in fila:
                copia_fila.append(celda)
            copia_planta.append(copia_fila)
        copia_mapa.append(copia_planta)

    #Sustituir los nodos visitados por *

    for i in visitados:
        pl,x,y = i
        casilla, _ = tipo_casilla(pl, x, y)
        if casilla not in MARCAS:
            copia_mapa[pl][x][y] = "*"

    #Sustituir todas las coordenadas por las que se ha pasado por =
    for i in acciones:
        pl,x,y = i
        casilla, _ = tipo_casilla(pl,x,y)
        if casilla not in MARCAS:
            copia_mapa[pl][x][y] = "="

    print("----- CAMINO RECORRIDO -----")
    plo, xo, yo = ORIGEN
    pld, xd, yd = DESTINO
    sal, _ = tipo_casilla(plo,xo,yo)

```

```

dest, _ = tipo_casilla(pld,xd,yd)

print(f"----- Salida: {sal} {ORIGEN} Destino: {dest} {DESTINO} -----")
for planta in copia_mapa:
    print()
    for i in planta:
        print(i)

def solucion_total(acciones,visitados,origen,destinos,costes,tiempos):

    copia_mapa = []

    for planta in plantas:
        copia_planta = []
        for fila in planta:
            copia_fila = []
            for celda in fila:
                copia_fila.append(celda)
            copia_planta.append(copia_fila)
        copia_mapa.append(copia_planta)

    #Sustituir los nodos visitados por *
    for i in visitados:
        pl,x,y = i
        casilla, _ = tipo_casilla(pl, x, y)
        if casilla not in MARCAS:
            copia_mapa[pl][x][y] = "*"

    #Sustituir todas las coordenadas por las que se ha pasado por =
    for i in acciones:
        pl,x,y = i
        casilla,_ = tipo_casilla(pl,x,y)
        if casilla not in MARCAS:
            copia_mapa[pl][x][y] = "="

    print("----- RESULTADO FINAL DE LA BUSQUEDA -----")

    print(f"El camino recorrido es: {acciones}")
    print(f"El coste total es de {sum(costes)}")
    print(f"La cantidad de nodos expandidos es de {len(visitados)}")
    print(f"El tiempo total de ejecucion es: {sum(tiempos)}")

    print("----- CAMINO RECORRIDO -----")

    print(f"----- Salida: {origen} Destinos: {destinos} -----")
    for planta in copia_mapa:
        print()
        for i in planta:
            print(i)

def comparativa(resultados, origen, destinos):
    #Enseñar el mapa final de todos los algoritmos
    for nombre, datos in resultados.items():
        print(f"----- El mapa resultante del algoritmo {nombre} -----")

```

```

solucion_total(
    datos["rutas"],
    datos["expandidos"],
    origen,
    destinos,
    datos["costes"],
    datos["tiempos"]
)

# Comparativa final de los algoritmos
print("----- COMPARATIVA FINAL DE LOS ALGORITMOS -----")
print("| Algoritmo | Tiempo Total (s) | Nodos Expandidos Totales | Coste Total |")
print("-----|-----|-----|-----|-----")

for nombre, datos in resultados.items():
    # Enseñar la suma los tiempos y la cantidad de nodos expandidos para todos los algoritmos
    tiempo_total = sum(datos["tiempos"])
    nodos_expandidos_totales = len(datos["expandidos"])
    coste_total = sum(datos["costes"])

# Resultados
print(f"| {nombre} | {tiempo_total} | {nodos_expandidos_totales} | {coste_total} |")

```

Orden de vista

Para BFS y DFS se usara el orden URDL (Up, Right, Down, Left):

En la búsqueda de sucesores utilizo el orden URDL (Up–Right–Down–Left). Este orden corresponde al movimiento en una matriz siguiendo el sistema habitual de coordenadas (filas y columnas) y garantiza que el algoritmo actúe de manera determinista, explorando siempre los vecinos en la misma secuencia. Además, usar un orden fijo permite que los diferentes algoritmos (BFS, DFS, Greedy y A*) sean comparables entre sí, evitando que la variación en el recorrido se deba al orden de expansión en vez de a la propia estrategia de búsqueda.

In [107...]

```

#Orden de insercion en la frontera: Up Right Down Left
DESPLAZAMIENTOS_URDL = [(-1,0), (0,1), (1,0), (0,-1)]


def acciones_disponibles(nodo):
    if debug:
        print("LOCALIZACION: ENTRANDO EN LA BUSQUEDA DE ACCIONES DISPONIBLES")
    pl, x, y = nodo.get_coordenadas()
    acciones_disponibles = []

    #Salto de planta si se encuentra en la casilla de ascensor, añadir la casilla
    #por heuristicas deberian ser mas atractivas las de la planta del destino
    #y para las no informadas añadirlas antes que las demas para que se priorice
    if plantas[pl][x][y] == 'E':

        for planta in range(PLANTAS):

            if planta != pl and plantas[planta][x][y] == 'E':


                acciones_disponibles.append(((planta, x, y), COST_ASC))

```

```
#Añadir una accion por cada casilla de movimiento disponible siempre que se
for disp_x,disp_y in DESPLAZAMIENTOS_URDL:

    nuevo_x = x + disp_x
    nuevo_y = y + disp_y

    #Se encuentra en el rango y no es una pared
    if 0 <= nuevo_x < FILAS and 0 <= nuevo_y < COLUMNAS:
        casilla, coste = tipo_casilla(pl,nuevo_x,nuevo_y)
        if casilla != "#":
            acciones_disponibles.append(((pl,nuevo_x,nuevo_y),coste))

    if debug:
        print(f"Las acciones disponibles del nodo de las coordenadas {nuevo_x},{nuevo_y} son: {acciones_disponibles}")

return acciones_disponibles
```

Heuristica manhattan:

La heurística Manhattan calcula la distancia mínima estimada desde un punto hasta el destino sumando las diferencias absolutas entre filas y columnas.

Se ha elegido Manhattan como heurística principal porque:

1. El movimiento permitido es únicamente ortogonal (URDL),
2. Manhattan es consistente, admisible y muy apropiada para un mapa de este estilo,
3. Es ideal cuando existen múltiples caminos equivalentes entre dos puntos.

El entorno multinivel, se ha pensado para contemplar el uso de ascensores: si el robot y el destino están en distinta planta, la heurística considera el mejor caso posible entre todos los ascensores disponibles (mínimo teórico), sumando el coste de subir.

Esta heurística no elige el ascensor, solo proporciona una cota inferior admisible. El algoritmo A* es quien decide realmente cuál es el mejor ascensor usando el coste real acumulado, permitiendo así el uso de múltiples ascensores

In [108...]

```
#Calculo de manhattan
def manhattan(ax, ay, bx, by):
    return abs(ax-bx) + abs(ay-by)

#heuristica de manhattan
def h_manhattan_estado(estado, destino):

    if debug:
        print("ENTRANDO EN EL CALCULO DE HEURISTICA MANHATTAN")

    pl, x, y = estado
    pld, xd, yd = destino

    #Si el objetivo esta en la misma planta se hace manhattan al destino unicamente
    if pl == pld:
        if debug:
            print("El objetivo esta en la misma planta")
        return manhattan(x, y, xd, yd)
```

```

if not ASCENSORES:
    ("ERROR: No hay ascensores mapeados")
    return 999**9

#Si el objetivo esta en una planta distinta se busca el mejor ascensor teori
#y se hace manhattan del estado al ascensor mas el del ascensor a el destino
mejor = 999**9
if debug:
    print(f"Estas en la planta {pl} y el objetivo en la planta {pld}")
for (xa, ya) in ASCENSORES:
    h = manhattan(x, y, xa, ya) + COST_ASC + manhattan(xa, ya, xd, yd)
    if h < mejor:
        mejor = h
if debug:
    print(f"La heuristicas del nodo {estado} es de {mejor}")
return mejor

```

Heuristica Euclidea:

La heurística Euclídea calcula la distancia en línea recta entre dos puntos usando el teorema de Pitágoras.

La euclídea mide la distancia directa, como si trazara una línea recta entre el nodo y el objetivo. Es la forma más "natural" de medir la proximidad en un plano.

In [109...]

```

#Calculo de distancia euclídea (Pitágoras)
def euclidea(ax, ay, bx, by):
    return ((ax - bx)**2 + (ay - by)**2) ** 0.5

def h_euclidea_estado(estado, destino):

    if debug:
        print("ENTRANDO EN EL CALCULO DE HEURISTICA EUCLIDEA")

    pl, x, y = estado
    pld, xd, yd = destino

    #Si el objetivo está en la misma planta, distancia euclídea directa
    if pl == pld:
        if debug:
            print("El objetivo está en la misma planta")
        return euclidea(x, y, xd, yd)

    #Si no hay ascensores, no se puede subir directamente
    if not ASCENSORES:
        print("ERROR: No hay ascensores mapeados")
        return 999**9

    #Si el objetivo esta en una planta distinta se busca el mejor ascensor teori
    #y se hace manhattan del estado al ascensor mas el del ascensor a el destino
    mejor = 999**9

    if debug:
        print(f"Estas en la planta {pl} y el objetivo en la planta {pld}")

    for (xa, ya) in ASCENSORES:
        h = euclidea(x, y, xa, ya) + COST_ASC + euclidea(xa, ya, xd, yd)
        if h < mejor:

```

```

        mejor = h

    if debug:
        print(f"La heurística del nodo {estado} es de {mejor}")

    return mejor

```

Expansion: Creacion de los nodos de todas las acciones disponibles

```

In [110...]:
def expand(frontier,nodo,heuristica,calculo_fn):
    # por cada acción disponible almacenada en el atributo actions
    for accion, coste in nodo.get_acciones_disponibles():

        pl,x,y = accion
        tipo, _ = tipo_casilla(pl,x,y)
        # instanciamos un nuevo nodo
        n = Nodo()
        n.set_estado(tipo)
        n.set_coordenadas(accion)
        n.set_padre(nodo)
        n.set_accion(nodo.get_accion()+[accion])
        n.set_acciones_disponibles(acciones_disponibles(n))

        n.set_coste(coste)
        n.set_coste_acumulado(nodo.get_coste_acumulado()+ n.get_coste())
        n.set_heuristica(heuristica(accion,DESTINO))
        n.set_fn(calculo_fn(n))
        n.set_profundidad(nodo.get_profundidad() + 1)
        # lo añadimos a la lista
        frontier.append(n)

    if mostrar_iteracion:
        print(n)

#Devolvemos lista
return frontier

```

A* Manhattan

```

In [111...]:
def graphsearch_a(origen,destino):

    frontier=[]
    visitados =[]
    found = False

    meapeado_inicial(origen,destino)

    n0 = Nodo()
    n0.set_estado(origen)
    n0.set_coordenadas(ORIGEN)
    n0.set_padre(None)
    n0.set_accion([ORIGEN])

    n0.set_acciones_disponibles(acciones_disponibles(n0))
    n0.set_coste(0)

```

```

n0.set_heuristica(h_manhattan_estado(ORIGEN,DESTINO))
n0.set_fn(fn_a(n0))
n0.set_profundidad(0)

print(n0)
frontier.append(n0)

#Comprobamos si el nodo inicial es el nodo objetivo
if n0.get_estado() == destino:
    found = True

while not found and frontier:

    #Ordenar la frontera por fn
    frontier.sort(key=clave_orden)
    if mostrar_iteracion:
        print("Frontera ordenada: ")
        imprimir_frontera(frontier)

    #extraemos elemento de la frontera según algoritmo
    n = frontier.pop(0)
    if mostrar_iteracion:
        print(f"Nuevo con f(n) menor es: {n.get_estado()}")

    # comprobamos si el nodo extraído es el objetivo,
    if n.get_estado() == destino:
        print("HAS LLEGADO AL OBJETIVO")
        found = True

    # en caso contrario lo expandimos llamando a la función expandir
    #Cambiar el calculo de fn para elegir algoritmo
    elif n.get_coordenadas() not in visitados:
        frontier = expand(frontier,n,h_manhattan_estado,fn_a)
        visitados.append(n.get_coordenadas())

    else:
        if mostrar_iteracion:
            print(f"El nodo {n.get_estado()} ya ha sido visitado por lo que no se ex")
        if not found and not frontier:
            print("EL RESULTADO NO PUEDE SER ENCONTRADO")
            return None,[]

return n, visitados

def treeseach_a(origen,destino):

    visitados = []
    frontier = []
    found = False

    meapeado_inicial(origen,destino)

```

```

n0 = Nodo()
n0.set_estado(origen)
n0.set_coordenadas(ORIGEN)
n0.set_padre(None)
n0.set_accion([ORIGEN])

n0.set_acciones_disponibles(acciones_disponibles(n0))
n0.set_coste(0)
n0.set_heuristica(h_manhattan_estado(ORIGEN,DESTINO))
n0.set_fn(fn_a(n0))
n0.set_profundidad(0)

print(n0)
frontier.append(n0)

#Comprobamos si el nodo inicial es el nodo objetivo
if n0.get_estado() == destino:
    found = True

while not found and frontier:

    #Ordenar la frontera por fn
    frontier.sort(key=clave_orden)
    if mostrar_iteracion:
        print("Frontera ordenada: ")
        imprimir_frontera(frontier)

    #extraemos elemento de la frontera según algoritmo
    n = frontier.pop(0)
    if mostrar_iteracion:
        print(f"Nuevo con f(n) menor es: {n.get_estado()}")

    # comprobamos si el nodo extraído es el objetivo,
    if n.get_estado() == destino:
        print("HAS LLEGADO AL OBJETIVO")
        found = True

    # en caso contrario lo expandimos llamando a la función expandir
    #Cambiar el calculo de fn para elegir algoritmo
    else:
        frontier = expand(frontier,n,h_manhattan_estado,fn_a)
        visitados.append(n.get_coordenadas())

    if not found and not frontier:
        print("EL RESULTADO NO PUEDE SER ENCONTRADO")
        return None, []

return n,visitados

```

Greedy Manhattan

In [112...]

```
def graphsearch_greedy(origen,destino):

    frontier=[ ]
    visitados =[ ]
    found = False

    meapeado_inicial(origen,destino)

    n0 = Nodo()
    n0.set_estado(origen)
    n0.set_coordenadas(ORIGEN)
    n0.set_padre(None)
    n0.set_accion([ORIGEN])

    n0.set_acciones_disponibles(acciones_disponibles(n0))
    n0.set_coste(0)
    n0.set_heuristica(h_manhattan_estado(ORIGEN,DESTINO))
    n0.set_fn(fn_a(n0))
    n0.set_profundidad(0)

    print(n0)
    frontier.append(n0)

#Comprobamos si el nodo inicial es el nodo objetivo
if n0.get_estado() == destino:
    found = True

while not found and frontier:

    #Ordenar la frontera por fn
    frontier.sort(key=clave_orden)
    if mostrar_iteracion:
        print("Frontera ordenada: ")
        imprimir_frontera(frontier)

    #extraemos elemento de la frontera según algoritmo
    n = frontier.pop(0)
    if mostrar_iteracion:
        print(f"Nuevo con f(n) menor es: {n.get_estado()}")

    # comprobamos si el nodo extraído es el objetivo,
    if n.get_estado() == destino:
        print("HAS LLEGADO AL OBJETIVO")
        found = True

    # en caso contrario lo expandimos llamando a la función expandir
    #Cambiar el calculo de fn para elegir algoritmo
    elif n.get_coordenadas() not in visitados:
        frontier = expand(frontier,n,h_manhattan_estado,fn_greedy)
```

```

    visitados.append(n.get_coordenadas())

else:
    if mostrar_iteracion:
        print(f"El nodo {n.get_estado()} ya ha sido visitado por lo que no se ex")
if not found and not frontier:
    print("EL RESULTADO NO PUEDE SER ENCONTRADO")
    return None, []

return n,visitados

def treesearch_greedy(origen,destino):

visitados = []
frontier= []
found = False

meapeado_inicial(origen,destino)

n0 = Nodo()
n0.set_estado(origen)
n0.set_coordenadas(ORIGEN)
n0.set_padre(None)
n0.set_accion([ORIGEN])

n0.set_acciones_disponibles(acciones_disponibles(n0))
n0.set_coste(0)
n0.set_heuristica(h_manhattan_estado(ORIGEN,DESTINO))
n0.set_fn(fn_a(n0))
n0.set_profundidad(0)

print(n0)
frontier.append(n0)

#Comprobamos si el nodo inicial es el nodo objetivo
if n0.get_estado() == destino:
    found = True

while not found and frontier:

    #Ordenar la frontera por fn
    frontier.sort(key=clave_orden)
    if mostrar_iteracion:
        print("Frontera ordenada: ")
        imprimir_frontera(frontier)

    #extraemos elemento de la frontera según algoritmo
    n = frontier.pop(0)
    if mostrar_iteracion:
        print(f"Nuevo con f(n) menor es: {n.get_estado()}")

    # comprobamos si el nodo extraido es el objetivo,

```

```

if n.get_estado() == destino:
    print("HAS LLEGADO AL OBJETIVO")
    found = True

# en caso contrario lo expandimos llamando a la función expandir
#Cambiar el calculo de fn para elegir algoritmo
else:
    frontier = expand(frontier,n,h_manhattan_estado,fn_greedy)
    visitados.append(n.get_coordenadas())

if not found and not frontier:
    print("EL RESULTADO NO PUEDE SER ENCONTRADO")
    return None, []

return n,visitados

```

A* Euclidea

In [113...]:

```

def graphsearch_a_euclidea(origen,destino):

    frontier=[]
    visitados = []
    found = False

    meapeado_inicial(origen,destino)

    n0 = Nodo()
    n0.set_estado(origen)
    n0.set_coordenadas(ORIGEN)
    n0.set_padre(None)
    n0.set_accion([ORIGEN])

    n0.set_acciones_disponibles(acciones_disponibles(n0))
    n0.set_coste(0)
    n0.set_heuristica(h_euclidea_estado(ORIGEN,DESTINO))
    n0.set_fn(fn_a(n0))
    n0.set_profundidad(0)

    print(n0)
    frontier.append(n0)

#Comprobamos si el nodo inicial es el nodo objetivo
if n0.get_estado() == destino:
    found = True

while not found and frontier:

    #Ordenar la frontera por fn
    frontier.sort(key=clave_orden)
    if mostrar_iteracion:

```

```

print("Frontera ordenada: ")
imprimir_frontera(frontier)

#extraemos elemento de la frontera según algoritmo
n = frontier.pop(0)
if mostrar_iteracion:
    print(f"Nodo con f(n) menor es: {n.get_estado()}")

# comprobamos si el nodo extraído es el objetivo,
if n.get_estado() == destino:
    print("HAS LLEGADO AL OBJETIVO")
    found = True

# en caso contrario lo expandimos llamando a la función expandir
#Cambiar el calculo de fn para elegir algoritmo
elif n.get_coordenadas() not in visitados:
    frontier = expand(frontier,n,h_euclidea_estado,fn_a)
    visitados.append(n.get_coordenadas())

else:
    if mostrar_iteracion:
        print(f"El nodo {n.get_estado()} ya ha sido visitado por lo que no se ex")
if not found and not frontier:
    print("EL RESULTADO NO PUEDE SER ENCONTRADO")
    return None, []

return n, visitados

def treesearch_a_euclidea(origen,destino):

    visitados = []
    frontier = []
    found = False

    meapeado_inicial(origen,destino)

    n0 = Nodo()
    n0.set_estado(origen)
    n0.set_coordenadas(ORIGEN)
    n0.set_padre(None)
    n0.set_accion([ORIGEN])

    n0.set_acciones_disponibles(acciones_disponibles(n0))
    n0.set_coste(0)
    n0.set_heuristica(h_euclidea_estado(ORIGEN,DESTINO))
    n0.set_fn(fn_a(n0))
    n0.set_profundidad(0)

    print(n0)
    frontier.append(n0)

```

```

#Comprobamos si el nodo inicial es el nodo objetivo
if n0.get_estado() == destino:
    found = True

while not found and frontier:

    #Ordenar la frontera por fn
    frontier.sort(key=clave_orden)
    if mostrar_iteracion:
        print("Frontera ordenada: ")
        imprimir_frontera(frontier)

    #extraemos elemento de la frontera según algoritmo
    n = frontier.pop(0)
    if mostrar_iteracion:
        print(f"Nodo con f(n) menor es: {n.get_estado()}")

    # comprobamos si el nodo extraído es el objetivo,
    if n.get_estado() == destino:
        print("HAS LLEGADO AL OBJETIVO")
        found = True

    # en caso contrario lo expandimos llamando a la función expandir
    #Cambiar el calculo de fn para elegir algoritmo
    else:
        frontier = expand(frontier,n,h_euclidea_estado,fn_a)
        visitados.append(n.get_coordenadas())

    if not found and not frontier:
        print("EL RESULTADO NO PUEDE SER ENCONTRADO")
        return None,[]

return n,visitados

```

Greedy Euclidea

In [114...]

```

def graphsearch_greedy_euclidea(origen,destino):

    frontier=[]
    visitados = []
    found = False

    meapeado_inicial(origen,destino)

    n0 = Nodo()
    n0.set_estado(origen)
    n0.set_coordenadas(ORIGEN)
    n0.set_padre(None)
    n0.set_accion([ORIGEN])

```

```

n0.set_acciones_disponibles(acciones_disponibles(n0))
n0.set_coste(0)
n0.set_heuristica(h_euclidea_estado(ORIGEN,DESTINO))
n0.set_fn(fn_a(n0))
n0.set_profundidad(0)

print(n0)
frontier.append(n0)

#Comprobamos si el nodo inicial es el nodo objetivo
if n0.get_estado() == destino:
    found = True

while not found and frontier:

    #Ordenar la frontera por fn
    frontier.sort(key=clave_orden)
    if mostrar_iteracion:
        print("Frontera ordenada: ")
        imprimir_frontera(frontier)

    #extraemos elemento de la frontera según algoritmo
    n = frontier.pop(0)
    if mostrar_iteracion:
        print(f"Nuevo con f(n) menor es: {n.get_estado()}")

    # comprobamos si el nodo extraído es el objetivo,
    if n.get_estado() == destino:
        print("HAS LLEGADO AL OBJETIVO")
        found = True

    # en caso contrario lo expandimos llamando a la función expandir
    #Cambiar el calculo de fn para elegir algoritmo
    elif n.get_coordenadas() not in visitados:
        frontier = expand(frontier,n,h_euclidea_estado,fn_greedy)
        visitados.append(n.get_coordenadas())

    else:
        if mostrar_iteracion:
            print(f"El nodo {n.get_estado()} ya ha sido visitado por lo que no se ex")
        if not found and not frontier:
            print("EL RESULTADO NO PUEDE SER ENCONTRADO")
            return None, []

return n,visitados

def treesearch_greedy_euclidea(origen,destino):

    visitados = []
    frontier = []

```

```

found = False

meapeado_inicial(origen,destino)

n0 = Nodo()
n0.set_estado(origen)
n0.set_coordenadas(ORIGEN)
n0.set_padre(None)
n0.set_accion([ORIGEN])

n0.set_acciones_disponibles(acciones_disponibles(n0))
n0.set_coste(0)
n0.set_heuristica(h_euclidea_estado(ORIGEN,DESTINO))
n0.set_fn(fn_a(n0))
n0.set_profundidad(0)

print(n0)
frontier.append(n0)

#Comprobamos si el nodo inicial es el nodo objetivo
if n0.get_estado() == destino:
    found = True

while not found and frontier:

    #Ordenar la frontera por fn

    frontier.sort(key=clave_orden)
    if mostrar_iteracion:
        print("Frontera ordenada: ")
        imprimir_frontera(frontier)

#extraemos elemento de la frontera según algoritmo
n = frontier.pop(0)
if mostrar_iteracion:
    print(f"Nuevo con f(n) menor es: {n.get_estado()}")

# comprobamos si el nodo extraído es el objetivo,
if n.get_estado() == destino:
    print("HAS LLEGADO AL OBJETIVO")
    found = True

# en caso contrario lo expandimos llamando a la función expandir
#Cambiar el calculo de fn para elegir algoritmo
else:
    frontier = expand(frontier,n,h_euclidea_estado,fn_greedy)
    visitados.append(n.get_coordenadas())

if not found and not frontier:
    print("EL RESULTADO NO PUEDE SER ENCONTRADO")
    return None, []

```

```
return n,visitados
```

BFS

In [115...]

```
def graphsearch_bfs(origen,destino):

    frontier=[]
    visitados = []
    found = False

    meapeado_inicial(origen,destino)

    n0 = Nodo()
    n0.set_estado(origen)
    n0.set_coordenadas(ORIGEN)
    n0.set_padre(None)
    n0.set_accion([ORIGEN])

    n0.set_acciones_disponibles(acciones_disponibles(n0))
    n0.set_coste(0)
    n0.set_heuristica(h_manhattan_estado(ORIGEN,DESTINO))
    n0.set_fn(fn_a(n0))
    n0.set_profundidad(0)

    print(n0)
    frontier.append(n0)

#Comprobamos si el nodo inicial es el nodo objetivo
if n0.get_estado() == destino:
    found = True

while not found and frontier:

    if mostrar_iteracion:
        print("Frontera:")
        imprimir_frontera(frontier)

    #No ordenar y sacar el primer elemento
#frontier.sort(key=clave_orden)

#extraemos elemento de la frontera según algoritmo
n = frontier.pop(0)
if mostrar_iteracion:
    print(f"Nuevo con f(n) menor es: {n.get_estado()}")

# comprobamos si el nodo extraído es el objetivo,
if n.get_estado() == destino:
    print("HAS LLEGADO AL OBJETIVO")
    found = True
```

```

# en caso contrario lo expandimos llamando a la función expandir
#Cambiar el calculo de fn para elegir algoritmo
elif n.get_coordenadas() not in visitados:
    frontier = expand(frontier,n,h_manhattan_estado,fn_greedy)
    visitados.append(n.get_coordenadas())


else:
    if mostrar_iteracion:
        print(f"El nodo {n.get_estado()} ya ha sido visitado por lo que no se ex")
    if not found and not frontier:
        print("EL RESULTADO NO PUEDE SER ENCONTRADO")
        return None, []

return n,visitados

def treesearch_bfs(origen,destino):

    visitados = []
    frontier=[]
    found = False

    meapeado_inicial(origen,destino)

    n0 = Nodo()
    n0.set_estado(origen)
    n0.set_coordenadas(ORIGEN)
    n0.set_padre(None)
    n0.set_accion([ORIGEN])

    n0.set_acciones_disponibles(acciones_disponibles(n0))
    n0.set_coste(0)
    n0.set_heuristica(h_manhattan_estado(ORIGEN,DESTINO))
    n0.set_fn(fn_a(n0))
    n0.set_profundidad(0)

    print(n0)
    frontier.append(n0)

#Comprobamos si el nodo inicial es el nodo objetivo
if n0.get_estado() == destino:
    found = True

while not found and frontier:

    #Ordenar la frontera por fn
    if mostrar_iteracion:
        print("Frontera:")
        imprimir_frontera(frontier)

#extraemos elemento de la frontera según algoritmo

```

```

n = frontier.pop(0)
if mostrar_iteracion:
    print(f"Nuevo nodo con f(n) menor es: {n.get_estado()}")


# comprobamos si el nodo extraído es el objetivo,
if n.get_estado() == destino:
    print("HAS LLEGADO AL OBJETIVO")
    found = True


# en caso contrario lo expandimos llamando a la función expandir
# Cambiar el cálculo de fn para elegir algoritmo
else:
    frontier = expand(frontier, n, h_manhattan_estado, fn_greedy)
    visitados.append(n.get_coordenadas())


if not found and not frontier or (n.get_profundidad() > PROFUNDIDAD_MAX) or (
    print("EL RESULTADO NO PUEDE SER ENCONTRADO")
    return None, []


return n, visitados

```

UCS

In [116...]

```

def graphsearch_dfs(origen, destino):

    frontier = []
    visitados = []
    found = False

    meapeado_inicial(origen, destino)

    n0 = Nodo()
    n0.set_estado(origen)
    n0.set_coordenadas(ORIGEN)
    n0.set_padre(None)
    n0.set_accion([ORIGEN])

    n0.set_acciones_disponibles(acciones_disponibles(n0))
    n0.set_coste(0)
    n0.set_heuristica(h_manhattan_estado(ORIGEN, DESTINO))
    n0.set_fn(fn_a(n0))
    n0.set_profundidad(0)

    print(n0)
    frontier.append(n0)


#Comprobamos si el nodo inicial es el nodo objetivo
if n0.get_estado() == destino:
    found = True

while not found and frontier:

```

```

if mostrar_iteracion:
    print("Frontera:")
    imprimir_frontera(frontier)

#No ordenar y sacar el primer elemento
#frontier.sort(key=clave_orden)

#extraemos elemento de la frontera según algoritmo
n = frontier.pop(-1)
if mostrar_iteracion:
    print(f"Nodo con f(n) menor es: {n.get_estado()}")

# comprobamos si el nodo extraído es el objetivo,
if n.get_estado() == destino:
    print("HAS LLEGADO AL OBJETIVO")
    found = True

# en caso contrario lo expandimos llamando a la función expandir
#Cambiar el calculo de fn para elegir algoritmo
elif n.get_coordenadas() not in visitados:
    frontier = expand(frontier,n,h_manhattan_estado,fn_greedy)
    visitados.append(n.get_coordenadas())

else:
    if mostrar_iteracion:
        print(f"El nodo {n.get_estado()} ya ha sido visitado por lo que no se ex")
    if not found and not frontier:
        print("EL RESULTADO NO PUEDE SER ENCONTRADO")
        return None, []

return n,visitados

def treesearch_dfs(origen,destino):

    visitados = []
    frontier = []
    found = False

    meapeado_inicial(origen,destino)

    n0 = Nodo()
    n0.set_estado(origen)
    n0.set_coordenadas(ORIGEN)
    n0.set_padre(None)
    n0.set_accion([ORIGEN])

    n0.set_acciones_disponibles(acciones_disponibles(n0))
    n0.set_coste(0)
    n0.set_heuristica(h_manhattan_estado(ORIGEN,DESTINO))
    n0.set_fn(fn_a(n0))
    n0.set_profundidad(0)

```

```

print(n0)
frontier.append(n0)

#Comprobamos si el nodo inicial es el nodo objetivo
if n0.get_estado() == destino:
    found = True

while not found and frontier:

    #Ordenar la frontera por fn
    if mostrar_iteracion:
        print("Frontera:")
        imprimir_frontera(frontier)

#extraemos elemento de la frontera según algoritmo
n = frontier.pop(-1)
if mostrar_iteracion:
    print(f"Nuevo con f(n) menor es: {n.get_estado()}")

# comprobamos si el nodo extraído es el objetivo,
if n.get_estado() == destino:
    print("HAS LLEGADO AL OBJETIVO")
    found = True

# en caso contrario lo expandimos llamando a la función expandir
#Cambiar el calculo de fn para elegir algoritmo
else:
    frontier = expand(frontier,n,h_manhattan_estado,fn_greedy)
    visitados.append(n.get_coordenadas())

    if not found and not frontier or (n.get_profundidad() > PROFUNDIDAD_MAX) or (
        print("EL RESULTADO NO PUEDE SER ENCONTRADO")
        return None, []

return n,visitados

```

Calculador de rutas multiples

Permite calcular la ruta pasando por multiples destinos

In [117...]

```

import time #Me ha dejado hugo
def ruta_multiple(origen,destinos,funcion_busqueda):

    origen_actual = origen
    rutas = []
    costes = []
    expandidos_totales = []
    tiempos = []
    nodos = []

    #Llamar a la funcion de busqueda elegida
    #por cada destino cambiando el origen al destino anterior

```

```

for destino in destinos:
    t0 = time.time()
    nodo_res, expandidos = funcion_busqueda(origen_actual,destino)
    t1 = time.time()
    tiempo = t1-t0

    if nodo_res == None:
        print(f"EL PROBLEMA NO TIENE SOLUCION: No se ha encontrado ruta desd
        return [],[],[],[]

    solucion(nodo_res,expandidos)

    nodos.append(nodo_res)
    rutas.extend(nodo_res.get_accion())
    costes.append(nodo_res.get_coste_acumulado())
    expandidos_totales.extend(expandidos)
    tiempos.append(tiempo)

    origen_actual = destino

solucion_total(rutas,expandidos_totales,origen,destinos,costes,tiempos)
return rutas,costes,expandidos_totales,tiempos

```

Analisis de todas las busquedas

Se usan todas las busquedas, mostrando sus resultados y comparandolos

```

In [118...]: def analisis(origen, destinos):

    #Diccionario con todos los algoritmos de busqueda creados
    algoritmos = {
        "A* Graph Manhattan": graphsearch_a,
        "A* Graph Euclidea": graphsearch_a_euclidea,
        "A* Tree Manhattan": treesearch_a,
        #Esta va lentisima si los prints estan activados, si da problemas poner
        "A* Tree Euclidea": treesearch_a_euclidea,
        "Greedy Graph Manhattan": graphsearch_greedy,
        "Greedy Graph Euclidea": graphsearch_greedy_euclidea,
        "Greedy Tree Manhattan": treesearch_greedy,
        "Greedy Tree Euclidea": treesearch_greedy_euclidea,
        "BFS Graph": graphsearch_bfs,
        "BFS Tree": treesearch_bfs,
        "DFS Graph": graphsearch_dfs,
        "DFS Tree": treesearch_dfs,
    }

    resultados = {}

    # Ejecutar cada algoritmo y guardar los resultados en el diccionario
    for nombre, funcion in algoritmos.items():
        rutas, costes, expandidos, tiempos = ruta_multiple(origen, destinos, fun
        resultados[nombre] = {
            "rutas": rutas,

```

```

    "costes": costes,
    "expandidos": expandidos,
    "tiempos": tiempos
}

comparativa(resultados,origen,destinos)

```

2) Caso de uso

Habrá varias formas posibles de visualizar los resultados:

1. Hacer la consulta directamente sobre los 2 puntos con el algoritmo deseado, que devolverá el nodo final resultante junto con la lista de todos los nodos visitados (Está bien para visualizar una ruta en específico y el proceso seguido con la frontera y los valores de los nodos creados, ya que de las otras formas es algo difícil ver uno en concreto):

```
res_graph_a,visitados_graph_a = graphsearch_a("Estacion C","Estanteria K")
```

2. Hacer la consulta para una lista de destinos y pasando el tipo de búsqueda a usar, esta imprimirá cada ruta individualmente y al final el resultado total junto con el mapa de todas las casillas recorridas y expandidas:

```
ruta_multiple("Estacion C",["Estanteria K","Estanteria H","Estanteria F"],graphsearch_a)
```

3. Hacer el análisis completo de todos los algoritmos de búsqueda para esa ruta, esta hará todo lo de las anteriores y al final imprimirá el mapa final de cada uno junto con una tabla con los resultados finales de cada algoritmo (Es la mejor para ver los resultados finales pero para ver cada caso en específico y el funcionamiento del algoritmo es algo molesto)

```
analisis("Estacion C",["Estanteria K","Estanteria H","Estanteria F"])
```

En caso de que un algoritmo de búsqueda entre en bucle o no tenga solución (esta programado de forma que si pasa de una profundidad configurable en las variables globales, devuelve que no tiene solución) saldrá por pantalla que no tiene solución y en las tablas los valores saldrán a 0

Funciones de forma individual (Usar para pruebas)

A*

```
In [119...]: #res_graph_a,visitados_graph_a = graphsearch_a("Estacion C","Estanteria K")
#solucion(res_graph_a,visitados_graph_a)
```

```
In [120...]: #res_tree_a,visitados_tree_a = treesearch_a("C","K")
#solucion(res_tree_a,visitados_tree_a)
```

```
In [121...]: #res_graph_a_euclidea , visitados_graph_euclidea = graphsearch_a_euclidea("C","K")
#solucion(res_graph_a_euclidea,visitados_graph_euclidea)
```

```
In [122...]: #res_tree_a_euclidea , visitados_tree_euclidea = treesearch_a_euclidea("C","K")
#solucion(res_tree_a_euclidea,visitados_tree_euclidea)
```

Greedy

```
In [123...]: #res_graph_greedy,visitados_graph_greedy = graphsearch_greedy("C","K")
#solucion(res_graph_greedy,visitados_graph_greedy)
```

```
In [124...]: #res_tree_greedy,visitados_tree_greedy = treesearch_greedy("C","K")
#solucion(res_tree_greedy,visitados_tree_greedy)
```

```
In [125...]: #res_graph_greedy_euclidea,visitados_graph_greedy_euclidea = graphsearch_greedy_euclidea("C","K")
#solucion(res_graph_greedy_euclidea,visitados_graph_greedy_euclidea)
```

```
In [126...]: #res_tree_greedy_euclidea,visitados_tree_greedy_euclidea = treesearch_greedy_euclidea("C","K")
#solucion(res_tree_greedy_euclidea,visitados_tree_greedy_euclidea)
```

BFS

```
In [127...]: #res_graph_bfs,visitados_graph_bfs = graphsearch_bfs("C","K")
#solucion(res_graph_bfs,visitados_graph_bfs)
```

```
In [128...]: #res_tree_bfs,visitados_tree_bfs = treesearch_bfs("C","K")
#solucion(res_tree_bfs,visitados_tree_bfs)
```

DFS

```
In [129...]: #res_graph_dfs,visitados_graph_dfs = graphsearch_dfs("C","K")
#solucion(res_graph_dfs,visitados_graph_dfs)
```

```
In [130...]: #Este ahora mismo entra en un bucle infinito
```

```
#res_tree_dfs,visitados_tree_dfs = treesearch_dfs("C","K")
#solucion(res_tree_dfs,visitados_tree_dfs)
```

Analisis completo

```
In [131...]: #ruta_multiple("C",["K","H","F"],treesearch_a_euclidea)
```

Caso real

Analisis de todos los algoritmos de busqueda

En caso de que se quede pillado es posible que sea por estar haciendo tantos prints, desactivar el print de la frontera y de los nodos expandidos poniendo a false la variable mostrar_iteraciones

Tambien modificar las variables siguientes como se necesite quitando el comentario y ejecutando el cuadrante

In [132...]

```
mostrar_iteracion = False #Activar/ Desactivar el imprimir la frontera y el nodo

#LIMITE_PRINT_FRONTERA = 50 #Limite de registros que se muestran de la frontera,
PROFUNDIDAD_MAX = 1000 #Profundidad maxima permitida antes de que lo marque como
FRONTERA_MAX = 10000
#debug = False # activar / descativar debug
```

Caso 1 : Recogida de paquete en la planta 0 y navegacion por 3 puntos en la planta 1

In [133...]

```
analisis("Estacion C",["Estanteria K","Estanteria H","Estanteria F"])
```

```

Nodo Estacion C(
    Estado: Estacion C
    Coordenadas: (0, 0, 0)
    Padre: None
    Acción: [(0, 0, 0)]
    Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 20
    F(n): 20
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria K(
    Estado: Estanteria K
    Coordenadas: (1, 9, 6)
    Padre: .
    Acción: [(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 1
    Coste Acumulado: 20
    Heuristica: 0
    F(n): 20
    Profundidad: 16
)
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]
El coste total es de 20
La cantidad de nodos expandidos es de 48
----- CAMINO RECORRIDO -----
----- Salida: Estacion C (0, 0, 0) Destino: Estanteria K (1, 9, 6) -----

[ '=', '=', '*', '**', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '**', '*', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '**', '*', '#', 'Estacion U', '.']
['*', '#', '**', '*', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '=', '=', 'Estanteria F', '*', '*', '*', '#', '.', '.']
[ '.', '*', '#', '=', '*', 'E', '*', '.', '#', '.']
[ '.', '*', 'T', '=', '#', '*', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '=', '*', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '=', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '=', '=', '=', '.', 'Estanteria L', '.']

Nodo Estanteria K(
    Estado: Estanteria K
)

```

```

    Coordenadas: (1, 9, 6)
    Padre: None
    Acción: [(1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 5
    F(n): 5
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 5
    Heuristica: 0
    F(n): 5
    Profundidad: 5
)
El camino recorrido es: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7),
(1, 6, 8)]
El coste total es de 5
La cantidad de nodos expandidos es de 9
----- CAMINO RECORRIDO -----
----- Salida: Estanteria K (1, 9, 6) Destino: Estanteria H (1, 6, 8) -----

[['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E'],
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.'],
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.'],
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.'],
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.'],
[ '.', '#', '.', '.', 'E', '.', '.', '#', '.'],
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.'],
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.'],
[ '.', '.', '.', '#', '.', '#', '.', '.', '.'],
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.'],

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E'],
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.'],
[ '.', '.', '#', '.', '.', '.', '#', '.', '.'],
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.'],
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.'],
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.'],
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.'],
[ '.', '#', '.', '.', '.', '.', '#', '=', 'Estanteria I', '.'],
[ '.', '.', '.', '#', '.', '#', '=', '=', '*', '.'],
[ '.', '.', '.', 'T', '.', '.', '=', '*', 'Estanteria L', '.']

Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
)

```

```

Coste Acumulado: 0
Heuristica: 7
F(n): 7
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 1
    Coste Acumulado: 7
    Heuristica: 0
    F(n): 7
    Profundidad: 7
)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 7
La cantidad de nodos expandidos es de 12
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estanteria F (1, 4, 3) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '*', '*', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 32
La cantidad de nodos expandidos es de 69
El tiempo total de ejecucion es: 0.005937099456787109
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria

```

F'] -----

```
['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '*', '*', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '*', '*', '*', '.', '#', 'Estacion U', '.']
['*', '#', '*', '*', '*', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
['.', '#', '.', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.']
['.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '*', '#', '=', '*', 'E', '=', '=', '#', '.', '.']
['.', '*', 'T', '=', '#', '*', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '=', '=', '*', '#', '=', 'Estanteria I', '.']
['.', '.', '.', '#', '=', '#', '=', '=', '*', '.']
['.', '.', '.', 'T', '=', '=', 'Estanteria K', '*', 'Estanteria L', '.']
```

Nodo Estacion C(

Estado: Estacion C

Coordenadas: (0, 0, 0)

Padre: None

Acción: [(0, 0, 0)]

Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 15.972527336075034

F(n): 15.972527336075034

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria K(

Estado: Estanteria K

Coordenadas: (1, 9, 6)

Padre: .

Acción: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]

Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],

Coste: 1

Coste Acumulado: 20

Heuristica: 0.0

F(n): 20.0

Profundidad: 16

)

El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]

El coste total es de 20

La cantidad de nodos expandidos es de 56

----- CAMINO RECORRIDO -----

----- Salida: Estacion C (0, 0, 0) Destino: Estanteria K (1, 9, 6) -----

```

['Estacion C', '*', '*', '*', '#', '#', '.', '.', '.', 'E']
['=', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['=', '=', '#', '*', '*', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '*', '*', '*', '.', '#', 'Estacion U', '.']
['*', '#', '*', '*', '*', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '*', '#', '.', '.', '.', '.', '#', '.', '.']
['*', 'E', '#', '#', '.', '#', '.', '.', '.']
['*', '=', '=', 'Estanteria F', '*', '*', '*', '#', '.', '.']
['*', '*', '#', '=', '*', 'E', '*', '.', '#', '.']
['*', '*', 'T', '=', '#', '*', 'G', '.', 'Estanteria H', '.']
['.', '#', '*', '=', '*', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '=', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '=', '=', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria K(
    Estado: Estanteria K
    Coordenadas: (1, 9, 6)
    Padre: None
    Acción: [(1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 3.605551275463989
    F(n): 3.605551275463989
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 5
    Heuristica: 0.0
    F(n): 5.0
    Profundidad: 5
)
El camino recorrido es: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8)]
El coste total es de 5
La cantidad de nodos expandidos es de 9
----- CAMINO RECORRIDO -----
----- Salida: Estanteria K (1, 9, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.']
['.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']

```

```
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '=', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '=', '=', '*', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '*', 'Estanteria L', '.']

Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 5.385164807134504
    F(n): 5.385164807134504
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 1
    Coste Acumulado: 7
    Heuristica: 0.0
    F(n): 7.0
    Profundidad: 7
)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
El coste total es de 7
La cantidad de nodos expandidos es de 14
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estanteria F (1, 4, 3) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
```

```
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '*', '*', '#', '.', '.']
[ '.', '.', '#', '*', '=', 'E', '*', '*', '#', '.']
[ '.', '.', 'T', '.', '#', '=', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '*', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----
```

El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]

El coste total es de 32

La cantidad de nodos expandidos es de 79

El tiempo total de ejecucion es: 0.005275249481201172

----- CAMINO RECORRIDO -----

----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria F'] -----

```
['Estacion C', '*', '*', '*', '#', '#', '.', '.', '.', 'E']
['=', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['=', '=', '#', '*', '*', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '*', '*', '*', '.', '#', 'Estacion U', '.']
['*', '#', '*', '*', '*', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

-----
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '*', '#', '.', '.', '.', '.', '#', '.', '.']
[ '*', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '*', '=', '=', 'Estanteria F', '=', '*', '*', '#', '.', '.']
[ '*', '*', '#', '=', 'E', '*', '*', '#', '.']
[ '*', '*', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '*', '=', '=', '*', '#', '=', 'Estanteria I', '.']
[ '.', '.', '.', '#', '=', '#', '=', '=', '*', '.']
[ '.', '.', '.', 'T', '=', '=', 'Estanteria K', '*', 'Estanteria L', '.']

-----
```

Nodo Estacion C(

Estado: Estacion C

Coordenadas: (0, 0, 0)

Padre: None

Acción: [(0, 0, 0)]

Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 20

F(n): 20

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria K(

Estado: Estanteria K

Coordenadas: (1, 9, 6)

Padre: .

Acción: [(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]

Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],

Coste: 1

Coste Acumulado: 20

Heuristica: 0

F(n): 20

Profundidad: 16

)

El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]

El coste total es de 20

La cantidad de nodos expandidos es de 391

----- CAMINO RECORRIDO -----

----- Salida: Estacion C (0, 0, 0) Destino: Estanteria K (1, 9, 6) -----

```
['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '*', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '*', '*', '*', '.', '#', 'Estacion U', '.']
['*', '#', '*', '*', '*', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
['.', '=', '=', 'Estanteria F', '*', '*', '*', '#', '.', '.']
['.', '*', '#', '=', '*', 'E', '*', '.', '#', '.']
['.', '*', 'T', '=', '#', '*', 'G', '.', 'Estanteria H', '.']
['.', '#', '.', '=', '=', '*', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '=', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '=', '=', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estanteria K(

Estado: Estanteria K

Coordenadas: (1, 9, 6)

Padre: None

Acción: [(1, 9, 6)]

Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 5

F(n): 5

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria H(

```

Estado: Estanteria H
Coordenadas: (1, 6, 8)
Padre: .
Acción: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8)]
Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
Coste: 1
Coste Acumulado: 5
Heuristica: 0
F(n): 5
Profundidad: 5
)
El camino recorrido es: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7),
(1, 6, 8)]
El coste total es de 5
La cantidad de nodos expandidos es de 18
----- CAMINO RECORRIDO -----
----- Salida: Estanteria K (1, 9, 6) Destino: Estanteria H (1, 6, 8) -----

[['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E'],
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.'],
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.'],
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.'],
['.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.'],
['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.'],
['.', '.', 'T', '.', '#', '.', '.', '.', '.'],
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.'],
['.', '.', '.', '#', '.', '.', '.', '.', '.'],
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.'],

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E'],
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.'],
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.'],
['.', 'E', '#', '#', '.', '#', '.', '.', '.'],
['.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.'],
['.', '.', '#', '.', '.', 'E', '.', '.', '#', '.'],
['.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.'],
['.', '#', '.', '.', '.', '#', '=', 'Estanteria I', '.'],
['.', '.', '.', '#', '.', '#', '=', '=', '*', '.'],
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '*', 'Estanteria L', '.']

Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 7
    F(n): 7
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1,
4, 4), (1, 4, 3)]

```

```

Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
Coste: 1
Coste Acumulado: 7
Heuristica: 0
F(n): 7
Profundidad: 7
)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6),
(1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 7
La cantidad de nodos expandidos es de 31
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estanteria F (1, 4, 3) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '*', '*', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 2, 1), (0, 3, 1),
(1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1,
7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8,
7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6),
(1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 32
La cantidad de nodos expandidos es de 440
El tiempo total de ejecucion es: 0.0874173641204834
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria
F'] ----

['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', 'E']
[*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
[*', '=', '#', '*', '*', 'Estanteria B', '.', 'T', '.']
[*', 'E', '#', '#', '*', '#', '.', '.', '.']
[*', '*', '*', '*', '*', '.', '#', 'Estacion U', '.']
[*', '#', '*', '*', '*', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']

```

```
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
['.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '*', '#', '=', '*', 'E', '=', '=', '#', '.']
['.', '*', 'T', '=', '#', '*', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '=', '=', '*', '#', '=', 'Estanteria I', '.']
['.', '.', '.', '#', '=', '#', '=', '=', '*', '.']
['.', '.', '.', 'T', '=', '=', 'Estanteria K', '*', 'Estanteria L', '.']
```

Nodo Estacion C(

Estado: Estacion C

Coordenadas: (0, 0, 0)

Padre: None

Acción: [(0, 0, 0)]

Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 15.972527336075034

F(n): 15.972527336075034

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria K(

Estado: Estanteria K

Coordenadas: (1, 9, 6)

Padre: .

Acción: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]

Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],

Coste: 1

Coste Acumulado: 20

Heuristica: 0.0

F(n): 20.0

Profundidad: 16

)

El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]

El coste total es de 20

La cantidad de nodos expandidos es de 1209

----- CAMINO RECORRIDO -----

----- Salida: Estacion C (0, 0, 0) Destino: Estanteria K (1, 9, 6) -----

```
['Estacion C', '*', '**', '**', '#', '#', '.', '.', '.', 'E']
[ '=', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
[ '=', '=', '#', '**', '**', 'Estanteria B', '.', 'T', '.']
[ '**', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
[ '**', '**', '**', '**', '.', '#', 'Estacion U', '.']
[ '**', '#', '**', '**', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.]
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '*', '#', '.', '.', '.', '#', '.', '.']
```

```
[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '*', '#', '.', '.', '.', '#', '.', '.']
```

```

[ '*', 'E', '#', '#', '.', '#', '.', '.', '.', '.', '.', '.']
[ '*', '=', '=', 'Estanteria F', '*', '*', '#', '.', '.', '.']
[ '*', '*', '#', '=', '*', 'E', '*', '.', '#', '.']
[ '*', '*', 'T', '=', '#', '*', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '*', '=', '=', '*', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '=', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '=', '=', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria K(
    Estado: Estanteria K
    Coordenadas: (1, 9, 6)
    Padre: None
    Acción: [(1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 3.605551275463989
    F(n): 3.605551275463989
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 5
    Heuristica: 0.0
    F(n): 5.0
    Profundidad: 5
)
El camino recorrido es: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8)]
El coste total es de 5
La cantidad de nodos expandidos es de 17
----- CAMINO RECORRIDO -----
----- Salida: Estanteria K (1, 9, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '=', 'Estanteria I', '.']

```

```

[ '.', '.', '.', '#', '.', '#', '=', '=', '*', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '*', 'Estanteria L', '.']
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 5.385164807134504
    F(n): 5.385164807134504
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 1
    Coste Acumulado: 7
    Heuristica: 0.0
    F(n): 7.0
    Profundidad: 7
)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
El coste total es de 7
La cantidad de nodos expandidos es de 32
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estanteria F (1, 4, 3) ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '*', '*', '#', '.', '.']
[ '.', '.', '#', '*', '=', 'E', '*', '*', '#', '.']
[ '.', '.', 'T', '.', '#', '=', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '*', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1),

```

```
(1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
```

El coste total es de 32

La cantidad de nodos expandidos es de 1258

El tiempo total de ejecucion es: 0.34572649002075195

----- CAMINO RECORRIDO -----

----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria F'] -----

```
['Estacion C', '*', '*', '*', '#', '#', '.', '.', '.', 'E']
['=', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['=', '=', '#', '*', '*', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '*', '*', '*', '.', '#', 'Estacion U', '.']
['*', '#', '*', '*', '*', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '*', '#', '.', '.', '.', '.', '#', '.', '.']
['*', 'E', '#', '#', '.', '#', '.', '.', '.']
['*', '=', '=', 'Estanteria F', '=', '*', '*', '#', '.', '.']
['*', '*', '#', '=', '=', 'E', '*', '*', '#', '.']
['*', '*', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '.']
['.', '#', '*', '=', '=', '*', '#', '=', 'Estanteria I', '.']
['.', '.', '.', '#', '=', '#', '=', '=', '*', '.']
['.', '.', '.', 'T', '=', '=', 'Estanteria K', '*', 'Estanteria L', '.']
```

Nodo Estacion C(

Estado: Estacion C

Coordenadas: (0, 0, 0)

Padre: None

Acción: [(0, 0, 0)]

Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 20

F(n): 20

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria K(

Estado: Estanteria K

Coordenadas: (1, 9, 6)

Padre: .

```
Acción: [(0, 0, 0), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3), (0, 1, 4), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 4, 5), (0, 5, 5), (1, 5, 5), (1, 5, 6), (1, 6, 6), (1, 6, 7), (1, 7, 7), (1, 8, 7), (1, 9, 7), (1, 9, 6)]
```

Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],

Coste: 1

Coste Acumulado: 23

Heuristica: 0

F(n): 0

Profundidad: 18

```

)
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3),
(0, 1, 4), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 4, 5), (0, 5, 5), (1, 5, 5), (1,
5, 6), (1, 6, 6), (1, 6, 7), (1, 7, 7), (1, 8, 7), (1, 9, 7), (1, 9, 6)]
El coste total es de 23
La cantidad de nodos expandidos es de 22
----- CAMINO RECORRIDO -----
----- Salida: Estacion C (0, 0, 0) Destino: Estanteria K (1, 9, 6) -----

['Estacion C', '=', '=', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '=', 'T', '*', '.', '#', '.', '.']
['.', '.', '#', '.', '=', '*', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
['.', '.', '.', '.', '=', '.', '#', 'Estacion U', '.']
['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', 'E', '=', '.', '#', '.']
['.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '*', '#', '=', 'Estanteria I', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '=', 'Estanteria L', '.']

Nodo Estanteria K(
    Estado: Estanteria K
    Coordenadas: (1, 9, 6)
    Padre: None
    Acción: [(1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 5
    F(n): 5
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 5
    Heuristica: 0
    F(n): 0
    Profundidad: 5
)
El camino recorrido es: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7),
(1, 6, 8)]
El coste total es de 5

```

La cantidad de nodos expandidos es de 5

----- CAMINO RECORRIDO -----

----- Salida: Estanteria K (1, 9, 6) Destino: Estanteria H (1, 6, 8) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '=', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '=', '=', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estanteria H(

Estado: Estanteria H

Coordenadas: (1, 6, 8)

Padre: None

Acción: [(1, 6, 8)]

Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 7

F(n): 7

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria F(

Estado: Estanteria F

Coordenadas: (1, 4, 3)

Padre: .

Acción: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]

Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],

Coste: 1

Coste Acumulado: 7

Heuristica: 0

F(n): 0

Profundidad: 7

)

El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]

El coste total es de 7

La cantidad de nodos expandidos es de 7

----- CAMINO RECORRIDO -----

----- Salida: Estanteria H (1, 6, 8) Destino: Estanteria F (1, 4, 3) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----
```

El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3), (0, 1, 4), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 4, 5), (0, 5, 5), (1, 5, 5), (1, 5, 6), (1, 6, 6), (1, 6, 7), (1, 7, 7), (1, 8, 7), (1, 9, 7), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]

El coste total es de 35

La cantidad de nodos expandidos es de 34

El tiempo total de ejecucion es: 0.0007979869842529297

----- CAMINO RECORRIDO -----

----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria F'] -----

```
['Estacion C', '=', '=', '=', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '=', 'T', '*', '.', '#', '.', '.']
[ '.', '.', '#', '.', '=', '*', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
[ '.', '.', '.', '=', '=', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- NODO ESTACION C -----
```

Nodo Estacion C(

Estado: Estacion C

Coordenadas: (0, 0, 0)

Padre: None

Acción: [(0, 0, 0)]

```

Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],
Coste: 0
Coste Acumulado: 0
Heuristica: 15.972527336075034
F(n): 15.972527336075034
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria K(
    Estado: Estanteria K
    Coordenadas: (1, 9, 6)
    Padre: .
    Acción: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 1
    Coste Acumulado: 20
    Heuristica: 0.0
    F(n): 0.0
    Profundidad: 16
)
El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]
El coste total es de 20
La cantidad de nodos expandidos es de 17
----- CAMINO RECORRIDO -----
----- Salida: Estacion C (0, 0, 0) Destino: Estanteria K (1, 9, 6) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
['=', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['=', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.']
['.', '=', '=', 'Estanteria F', '.', '.', '.', '#', '.', '.']
['.', '.', '#', '=', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '=', '#', '.', 'G', '.', 'Estanteria H', '.']
['.', '#', '.', '=', '=', '*', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '=', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '=', '=', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria K(
    Estado: Estanteria K
    Coordenadas: (1, 9, 6)
    Padre: None
    Acción: [(1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 0

```

```

Coste Acumulado: 0
Heuristica: 3.605551275463989
F(n): 3.605551275463989
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 5
    Heuristica: 0.0
    F(n): 0.0
    Profundidad: 5
)
El camino recorrido es: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7),
(1, 6, 8)]
El coste total es de 5
La cantidad de nodos expandidos es de 5
----- CAMINO RECORRIDO -----
----- Salida: Estanteria K (1, 9, 6) Destino: Estanteria H (1, 6, 8) -----

[['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']]
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '=', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '=', '=', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 5.385164807134504
    F(n): 5.385164807134504
    Profundidad: 0
)

```

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria F(

Estado: Estanteria F

Coordenadas: (1, 4, 3)

Padre: .

Acción: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]

Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],

Coste: 1

Coste Acumulado: 7

Heuristica: 0.0

F(n): 0.0

Profundidad: 7

)

El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]

El coste total es de 7

La cantidad de nodos expandidos es de 7

----- CAMINO RECORRIDO -----

----- Salida: Estanteria H (1, 6, 8) Destino: Estanteria F (1, 4, 3) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']
```

```
[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '=', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]

El coste total es de 32

La cantidad de nodos expandidos es de 29

El tiempo total de ejecucion es: 0.0007157325744628906

----- CAMINO RECORRIDO -----

----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria F'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '=', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '=', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
```

```
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '=', '=', 'Estanteria F', '=', '.', '.', '#', '.', '.']
[ '.', '.', '#', '=', 'E', '.', '.', '#', '.', '.']
[ '.', '.', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '=', '*', '#', '=', 'Estanteria I', '.']
[ '.', '.', '.', '#', '=', '#', '=', '=', '.', '.']
[ '.', '.', '.', 'T', '=', '=', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion C(
    Estado: Estacion C
    Coordenadas: (0, 0, 0)
    Padre: None
    Acción: [(0, 0, 0)]
    Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 20
    F(n): 20
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria K(
    Estado: Estanteria K
    Coordenadas: (1, 9, 6)
    Padre: .
    Acción: [(0, 0, 0), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3), (0, 1, 4), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 4, 5), (0, 5, 5), (1, 5, 5), (1, 5, 6), (1, 6, 6), (1, 6, 7), (1, 7, 7), (1, 8, 7), (1, 9, 7), (1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 1
    Coste Acumulado: 23
    Heuristica: 0
    F(n): 0
    Profundidad: 18
)
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3), (0, 1, 4), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 4, 5), (0, 5, 5), (1, 5, 5), (1, 5, 6), (1, 6, 6), (1, 6, 7), (1, 7, 7), (1, 8, 7), (1, 9, 7), (1, 9, 6)]
El coste total es de 23
La cantidad de nodos expandidos es de 26
----- CAMINO RECORRIDO -----
----- Salida: Estacion C (0, 0, 0) Destino: Estanteria K (1, 9, 6) -----

['Estacion C', '=', '=', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '=', 'T', '*', '.', '#', '.', '.']
[ '.', '.', '#', '.', '=', '*', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
[ '.', '.', '.', '=', '=', '.', '#', 'Estacion U', '.']
```

```
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '*', '#', '=', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '=', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '=', 'Estanteria L', '.']

Nodo Estanteria K(
    Estado: Estanteria K
    Coordenadas: (1, 9, 6)
    Padre: None
    Acción: [(1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 5
    F(n): 5
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 5
    Heuristica: 0
    F(n): 0
    Profundidad: 5
)
El camino recorrido es: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8)]
El coste total es de 5
La cantidad de nodos expandidos es de 5
----- CAMINO RECORRIDO -----
----- Salida: Estanteria K (1, 9, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '=', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '=', '=', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 7
    F(n): 7
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1,
4, 4), (1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 1
    Coste Acumulado: 7
    Heuristica: 0
    F(n): 0
    Profundidad: 7
)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6),
(1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 7
La cantidad de nodos expandidos es de 7
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estanteria F (1, 4, 3) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '#', '.', '.']
```

```
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
['.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3),
(0, 1, 4), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 4, 5), (0, 5, 5), (1, 5, 5), (1,
5, 6), (1, 6, 6), (1, 6, 7), (1, 7, 7), (1, 8, 7), (1, 9, 7), (1, 9, 6), (1, 9,
6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7),
(1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 35
La cantidad de nodos expandidos es de 38
El tiempo total de ejecucion es: 0.0008671283721923828
```

```
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria
F'] -----
```

```
['Estacion C', '=', '=', '=', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '=', 'T', '*', '.', '#', '.', '.']
['.', '.', '#', '.', '=', '*', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
['.', '.', '.', '.', '=', '.', '#', 'Estacion U', '.']
['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
['.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '*', '#', '=', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '=', '=', '.', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '=', 'Estanteria L', '.']
```

Nodo Estacion C(

Estado: Estacion C

Coordenadas: (0, 0, 0)

Padre: None

Acción: [(0, 0, 0)]

Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 15.972527336075034

F(n): 15.972527336075034

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria K(

Estado: Estanteria K

Coordenadas: (1, 9, 6)

Padre: .

```

    Acción: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1,
4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8,
4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 1
    Coste Acumulado: 20
    Heuristica: 0.0
    F(n): 0.0
    Profundidad: 16
)
El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1),
(1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1,
7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]
El coste total es de 20
La cantidad de nodos expandidos es de 17
----- CAMINO RECORRIDO -----
----- Salida: Estacion C (0, 0, 0) Destino: Estanteria K (1, 9, 6) -----

[['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E'],
['=', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.'],
['=', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.'],
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.'],
['.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.'],
['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.'],
['.', '.', 'T', '.', '#', '.', '.', '.', '.'],
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.'],
['.', '.', '.', '#', '.', '.', '.', '.', '.'],
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.'],

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E'],
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.'],
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.'],
['.', 'E', '#', '#', '.', '#', '.', '.', '.'],
['.', '=', '=', 'Estanteria F', '.', '.', '.', '#', '.', '.'],
['.', '.', '#', '=', '.', 'E', '.', '.', '#', '.'],
['.', '.', 'T', '=', '#', '.', 'G', '.', 'Estanteria H', '.'],
['.', '#', '.', '=', '=', '#', '.', 'Estanteria I', '.'],
['.', '.', '.', '#', '=', '#', '.', '.', '.', '.'],
['.', '.', '.', 'T', '=', '=', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria K(
    Estado: Estanteria K
    Coordenadas: (1, 9, 6)
    Padre: None
    Acción: [(1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 3.605551275463989
    F(n): 3.605551275463989
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
```

```

Coste: 1
Coste Acumulado: 5
Heuristica: 0.0
F(n): 0.0
Profundidad: 5
)
El camino recorrido es: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7),
(1, 6, 8)]
El coste total es de 5
La cantidad de nodos expandidos es de 5
----- CAMINO RECORRIDO -----
----- Salida: Estanteria K (1, 9, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '=', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '=', '=', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 5.385164807134504
    F(n): 5.385164807134504
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 5, 5), (1, 5, 4), (1,
4, 4), (1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 1
    Coste Acumulado: 7
    Heuristica: 0.0
    F(n): 0.0
)

```

```

Profundidad: 7
)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 5, 5),
(1, 5, 4), (1, 4, 4), (1, 4, 3)]
El coste total es de 7
La cantidad de nodos expandidos es de 7
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estanteria F (1, 4, 3) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '=', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1),
(1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1,
7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8,
7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5),
(1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
El coste total es de 32
La cantidad de nodos expandidos es de 29
El tiempo total de ejecucion es: 0.0006978511810302734
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria
F'] ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '=', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '=', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '=', 'Estanteria F', '=', '.', '.', '#', '.', '.']
[ '.', '.', '#', '=', 'E', '.', '.', '#', '.']
```

```

[ '.', '.', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '=', '=', '*', '#', '=', 'Estanteria I', '.']
[ '.', '.', '.', '#', '=', '#', '=', '=', '.', '.']
[ '.', '.', '.', 'T', '=', '=', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion C(
    Estado: Estacion C
    Coordenadas: (0, 0, 0)
    Padre: None
    Acción: [(0, 0, 0)]
    Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 20
    F(n): 20
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria K(
    Estado: Estanteria K
    Coordenadas: (1, 9, 6)
    Padre: .
    Acción: [(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 1
    Coste Acumulado: 20
    Heuristica: 0
    F(n): 0
    Profundidad: 16
)
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6)]
El coste total es de 20
La cantidad de nodos expandidos es de 159
----- CAMINO RECORRIDO -----
----- Salida: Estacion C (0, 0, 0) Destino: Estanteria K (1, 9, 6) -----

['Estacion C', '=', '*', '**', '#', '#', '*', '*', '*', 'E']
['*', 'Estanteria A', '#', '*', 'T', '*', '*', '#', '*', '*']
['*', '=', '#', '*', '**', 'Estanteria B', '*', 'T', '*']
['*', 'E', '#', '#', '*', '#', '**', '**', '**']
['*', '**', '**', '**', '**', '#', 'Estacion U', '**']
['*', '#', '**', '**', 'E', '**', '**', '#', '**']
['*', '**', 'T', '**', '#', '**', '**', '**', '**']
['*', '#', '**', 'Estacion R', '**', '**', '#', '**', 'Estanteria X', '**']
['*', '**', '**', '#', '**', '**', '**', '**', '.']
['*', '**', 'T', '**', '**', '**', '**', '.', '.']

['*', '**', '**', '**', '**', 'Estanteria J', '**', '**', 'E']
['*', 'Estanteria D', '**', 'T', '**', '**', '**', '**', '**']
['*', '**', '#', '**', '**', '**', '#', '**', '**']
['*', 'E', '#', '#', '**', '**', '**', '**', '**']
['*', '=', '=', 'Estanteria F', '**', '**', '**', '#', '**', '**']
['*', '**', '#', '=', '**', 'E', '**', '**', '#', '.']
['*', '**', 'T', '=', '#', '**', 'G', '**', 'Estanteria H', '**']
['*', '#', '**', '=', '**', '#', '**', 'Estanteria I', '.']

```

```

['*', '*', '*', '#', '=', '#', '.', '*', '.', '.']
['*', '*', '*', 'T', '=', '=', 'Estanteria K', '.', 'Estanteria L', '.']
Nodo Estanteria K(
    Estado: Estanteria K
    Coordenadas: (1, 9, 6)
    Padre: None
    Acción: [(1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 5
    F(n): 5
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 5
    Heuristica: 0
    F(n): 0
    Profundidad: 5
)
El camino recorrido es: [(1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7),
(1, 6, 8)]
El coste total es de 5
La cantidad de nodos expandidos es de 18
----- CAMINO RECORRIDO -----
----- Salida: Estanteria K (1, 9, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '*', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '*', '.', '#', '=', 'Estanteria I', '.']
[ '.', '.', '#', '*', '#', '=', '=', '*', '*']
[ '.', '.', '*', 'T', '*', '*', 'Estanteria K', '*', 'Estanteria L', '*']
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)

```

```

Padre: None
Acción: [(1, 6, 8)]
Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
Coste: 0
Coste Acumulado: 0
Heuristica: 7
F(n): 7
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 1
    Coste Acumulado: 7
    Heuristica: 0
    F(n): 0
    Profundidad: 7
)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 7
La cantidad de nodos expandidos es de 51
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estanteria F (1, 4, 3) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '*', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '*', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '*', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '*', '.', '*', '*']
[ '.', '.', '#', '.', '.', '*', '*', '#', '*', '*']
[ '.', 'E', '#', '#', '*', '#', '*', '*', '*', '*']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '*', '*']
[ '.', '.', '#', '*', '*', 'E', '=', '=', '#', '*']
[ '.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '*']
[ '.', '#', '.', '*', '*', '#', '*', 'Estanteria I', '*']
[ '.', '.', '.', '#', '*', '#', '*', '*', '*', '*']
[ '.', '.', '.', 'T', '*', '*', 'Estanteria K', '*', 'Estanteria L', '*']
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 2, 1), (0, 3, 1), (1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1, 7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 32

```

La cantidad de nodos expandidos es de 228
 El tiempo total de ejecucion es: 0.00873708724975586

----- CAMINO RECORRIDO -----

----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria F'] -----

```
['Estacion C', '=', '*', '#', '#', '*', '*', '*', 'E']
['*', 'Estanteria A', '#', '*', 'T', '*', '*', '#', '*', '*']
['*', '=', '#', '*', '*', '*', 'Estanteria B', '*', 'T', '*']
['*', 'E', '#', '#', '*', '#', '*', '*', '*', '*']
['*', '*', '*', '*', '*', '*', '#', 'Estacion U', '*']
['*', '#', '*', '*', '*', 'E', '*', '*', '#', '*']
['*', '*', 'T', '*', '#', '*', '*', '*', '*', '*']
['*', '#', '*', 'Estacion R', '*', '*', '#', '*', 'Estanteria X', '*']
['*', '*', '#', '*', '#', '*', '*', '*', '*', '.']
['*', '*', 'T', '*', '*', '*', '*', '.', '.']

['*', '*', '*', '*', '*', '*', 'Estanteria J', '*', '*', 'E']
['*', 'Estanteria D', '*', 'T', '*', '*', '*', '*', '*', '*']
['*', '*', '#', '*', '*', '*', '#', '*', '*']
['*', 'E', '#', '#', '*', '#', '*', '*', '*']
['*', '=', '=', 'Estanteria F', '=', '=', '=', '#', '*', '*']
['*', '#', '=', '*', 'E', '=', '=', '#', '*']
['*', '*', 'T', '=', '#', '*', 'G', '=', 'Estanteria H', '*']
['*', '#', '*', '=', '=', '*', '#', '=', 'Estanteria I', '*']
['*', '*', '#', '=', '#', '=', '=', '*', '*']
['*', '*', 'T', '=', '=', 'Estanteria K', '*', 'Estanteria L', '*']
```

Nodo Estacion C(

- Estado: Estacion C
- Coordenadas: (0, 0, 0)
- Padre: None
- Acción: [(0, 0, 0)]
- Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)], Coste: 0
- Coste Acumulado: 0
- Heuristica: 20
- F(n): 20
- Profundidad: 0

)

EL RESULTADO NO PUEDE SER ENCONTRADO

EL PROBLEMA NO TIENE SOLUCION: No se ha encontrado ruta desde Estacion C a Estanteria K

Nodo Estacion C(

- Estado: Estacion C
- Coordenadas: (0, 0, 0)
- Padre: None
- Acción: [(0, 0, 0)]
- Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)], Coste: 0
- Coste Acumulado: 0
- Heuristica: 20
- F(n): 20
- Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria K(

- Estado: Estanteria K
- Coordenadas: (1, 9, 6)

Padre: .

Acción: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 3, 0), (0, 4, 0), (0, 5, 0), (0, 6, 0), (0, 7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 2), (0, 6, 2), (0, 6, 3), (0, 5, 3), (0, 5, 2), (0, 4, 2), (0, 4, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3), (0, 2, 3), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 5, 4), (0, 5, 5), (0, 5, 6), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2, 9), (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6), (1, 0, 5), (1, 0, 4), (1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6)]

Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],

Coste: 1

Coste Acumulado: 106

Heuristica: 0

F(n): 0

Profundidad: 98

)

El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 3, 0), (0, 4, 0), (0, 5, 0), (0, 6, 0), (0, 7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 2), (0, 6, 2), (0, 6, 3), (0, 5, 3), (0, 5, 2), (0, 4, 2), (0, 4, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3), (0, 2, 3), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 5, 4), (0, 5, 5), (0, 5, 6), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2, 9), (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6), (1, 0, 5), (1, 0, 4), (1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6)]

El coste total es de 106

La cantidad de nodos expandidos es de 107

----- CAMINO RECORRIDO -----

----- Salida: Estacion C (0, 0, 0) Destino: Estanteria K (1, 9, 6) -----

```
['Estacion C', '=', '=', '=', '#', '#', '=', '=', '=', 'E']
[ '=', 'Estanteria A', '#', '=', 'T', '=', '=', '#', '=', '=']
[ '=', '=', '#', '=', '=', '=', 'Estanteria B', '=', 'T', '=']
[ '=', 'E', '#', '#', '=', '#', '=', '=', '=', '=']
[ '=', '=', '*', '=', '*', '=', '#', 'Estacion U', '=']
[ '=', '#', '=', '=', '=', 'E', '=', '*', '#', '*']
[ '=', '*', 'T', '=', '#', '=', '=', '=', '=', '=']
[ '=', '#', '=', 'Estacion R', '=', '=', '#', '=', 'Estanteria X', '=']
[ '=', '*', '*', '#', '.', '#', '*', '=', '=', '=']
[ '=', '=', 'T', '=', '=', '=', '=', '=', '=', '=']

[ '=', '=', '=', '=', '=', 'Estanteria J', '=', '=', 'E']
[ '=', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '=', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '=', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '=', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '=', '.', '.', '#', '.', '.', '.', '#', '.']
[ '=', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '=', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
```

```

[ '=', '.', '.', '#', '.', '#', '.', '.', '.', '.', '.']
[ '=', '=', '=', 'T', '=', '=', 'Estanteria K', '.', 'Estanteria L', '.']
Nodo Estanteria K(
    Estado: Estanteria K
    Coordenadas: (1, 9, 6)
    Padre: None
    Acción: [(1, 9, 6)]
    Acciones disponibles: [((1, 8, 6), 1), ((1, 9, 7), 1), ((1, 9, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 5
    F(n): 5
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 9, 6), (1, 9, 5), (1, 9, 4), (1, 9, 3), (1, 9, 2), (1, 9, 1), (1, 9, 0), (1, 8, 0), (1, 8, 1), (1, 8, 2), (1, 7, 2), (1, 7, 3), (1, 7, 4), (1, 7, 5), (1, 6, 5), (1, 6, 6), (1, 6, 7), (1, 7, 7), (1, 8, 7), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 28
    Heuristica: 0
    F(n): 0
    Profundidad: 27
)
El camino recorrido es: [(1, 9, 6), (1, 9, 5), (1, 9, 4), (1, 9, 3), (1, 9, 2), (1, 9, 1), (1, 9, 0), (1, 8, 0), (1, 8, 1), (1, 8, 2), (1, 7, 2), (1, 7, 3), (1, 7, 4), (1, 7, 5), (1, 6, 5), (1, 6, 6), (1, 6, 7), (1, 7, 7), (1, 8, 7), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 6, 8)]
El coste total es de 28
La cantidad de nodos expandidos es de 29
----- CAMINO RECORRIDO -----
----- Salida: Estanteria K (1, 9, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.', '.']
[ '.', '.', 'T', '.', '#', '=', 'G', '=', 'Estanteria H', '=']

```

```
[ '.', '#', '=', '=', '=', '#', '=', 'Estanteria I', '=']
[ '=', '=', '=', '#', '*', '#', '*', '=', '=', '=']
[ '=', '=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=']
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 7
    F(n): 7
    Profundidad: 0
)
```

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria F(

```
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7, 3), (1, 7, 2), (1, 8, 2), (1, 8, 1), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 5, 9), (1, 4, 9), (1, 4, 8), (1, 3, 8), (1, 3, 7), (1, 3, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 1
    Coste Acumulado: 38
    Heuristica: 0
    F(n): 0
    Profundidad: 37
)
```

El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7, 3), (1, 7, 2), (1, 8, 2), (1, 8, 1), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 5, 9), (1, 4, 9), (1, 4, 8), (1, 3, 8), (1, 3, 7), (1, 3, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]

El coste total es de 38

La cantidad de nodos expandidos es de 38

----- CAMINO RECORRIDO -----

----- Salida: Estanteria H (1, 6, 8) Destino: Estanteria F (1, 4, 3) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '#', '.', '.']
```

```
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '=', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '=', '=']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '=']
[ '.', '.', 'T', '.', '#', '=', 'G', '=', 'Estanteria H', '=']
[ '.', '#', '=', '=', '=', '=', '#', '=', 'Estanteria I', '=']
[ '=', '=', '#', '.', '#', '#', '*', '=', '=', '=']
[ '=', '=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=']
----- RESULTADO FINAL DE LA BUSQUEDA -----
```

El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 3, 0), (0, 4, 0), (0, 5, 0), (0, 6, 0), (0, 7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 2), (0, 6, 2), (0, 6, 3), (0, 5, 3), (0, 5, 2), (0, 4, 2), (0, 4, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3), (0, 2, 3), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 5, 4), (0, 5, 5), (0, 5, 6), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2, 9), (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6), (1, 0, 5), (1, 0, 4), (1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 9, 5), (1, 9, 4), (1, 9, 3), (1, 9, 2), (1, 9, 1), (1, 9, 0), (1, 8, 0), (1, 8, 1), (1, 8, 2), (1, 7, 2), (1, 7, 3), (1, 7, 4), (1, 7, 5), (1, 6, 5), (1, 6, 6), (1, 6, 7), (1, 7, 7), (1, 8, 7), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 7, 8), (1, 6, 9), (1, 5, 9), (1, 4, 9), (1, 4, 8), (1, 3, 8), (1, 3, 7), (1, 3, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]

El coste total es de 172

La cantidad de nodos expandidos es de 174

El tiempo total de ejecucion es: 0.004993915557861328

----- CAMINO RECORRIDO -----

----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria F'] -----

```
['Estacion C', '=', '=', '=', '#', '#', '=', '=', '=', 'E']
[ '=', 'Estanteria A', '#', '=', 'T', '=', '=', '#', '=', '=']
[ '=', '#', '=', '=', '=', 'Estanteria B', '=', 'T', '=']
[ '=', 'E', '#', '#', '=', '#', '=', '=', '=', '=']
[ '=', '=', '*', '=', '*', '=', '#', 'Estacion U', '=']
[ '=', '#', '=', '=', '=', 'E', '=', '*', '#', '*']
[ '=', '*', 'T', '=', '#', '=', '=', '=', '=', '=']
[ '=', '#', '=', 'Estacion R', '=', '=', '#', '=', 'Estanteria X', '=']
[ '=', '*', '*', '#', '.', '#', '*', '=', '=', '=']
[ '=', '=', 'T', '=', '=', '=', '=', '=', '=', '=']

[ '=', '=', '=', '=', '=', 'Estanteria J', '=', '=', 'E']
[ '=', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '=', 'E', '#', '#', '#', '=', '=', '=', '=', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '=', '=']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '=']
[ '.', '.', 'T', '.', '#', '=', 'G', '=', 'Estanteria H', '=']
[ '=', '#', '=', '=', '=', '#', '=', 'Estanteria I', '=']
[ '=', '=', '#', '*', '#', '*', '=', '=', '=']
[ '=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=']
```

Nodo Estacion C(

```

Estado: Estacion C
Coordenadas: (0, 0, 0)
Padre: None
Acción: [(0, 0, 0)]
Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],
Coste: 0
Coste Acumulado: 0
Heuristica: 20
F(n): 20
Profundidad: 0
)

EL RESULTADO NO PUEDE SER ENCONTRADO
EL PROBLEMA NO TIENE SOLUCION: No se ha encontrado ruta desde Estacion C a Estanteria K
----- El mapa resultante del algoritmo A* Graph Manhattan es -----
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 2, 1), (0, 3, 1),
(1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1,
7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8,
7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6),
(1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 32
La cantidad de nodos expandidos es de 69
El tiempo total de ejecucion es: 0.005937099456787109
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria F'] ----

['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '*', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '*', '*', '*', '.', '#', 'Estacion U', '.']
['*', '#', '*', '*', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '#', '.', '.', '.', '.']
['.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '*', '#', '=', '*', 'E', '=', '=', '#', '.']
['.', '*', 'T', '=', '#', '*', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '=', '=', '*', '#', '=', 'Estanteria I', '.']
['.', '.', '.', '#', '=', '#', '=', '=', '*', '.']
['.', '.', '.', 'T', '=', '=', 'Estanteria K', '*', 'Estanteria L', '.']

----- El mapa resultante del algoritmo A* Graph Euclidea es -----
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1),
(1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1,
7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8,
7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5),
(1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
El coste total es de 32
La cantidad de nodos expandidos es de 79

```

```

El tiempo total de ejecucion es: 0.005275249481201172
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria F'] ----

['Estacion C', '*', '*', '*', '#', '#', '.', '.', '.', 'E']
['=', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['=', '=', '#', '*', '*', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '*', '*', '*', '.', '#', 'Estacion U', '.']
['*', '#', '*', '*', '*', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '*', '#', '.', '.', '.', '.', '#', '.', '.']
['*', 'E', '#', '#', '#', '.', '.', '.', '.']
['*', '=', '=', 'Estanteria F', '=', '*', '*', '#', '.', '.']
['*', '*', '#', '=', '=', 'E', '*', '*', '#', '.']
['*', '*', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '.']
['.', '#', '*', '=', '=', '*', '#', '=', 'Estanteria I', '.']
['.', '.', '.', '#', '=', '#', '=', '=', '*', '.']
['.', '.', '.', 'T', '=', '=', 'Estanteria K', '*', 'Estanteria L', '.']

----- El mapa resultante del algoritmo A* Tree Manhattan es -----
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 2, 1), (0, 3, 1),
(1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1,
7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8,
7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6),
(1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 32
La cantidad de nodos expandidos es de 440
El tiempo total de ejecucion es: 0.0874173641204834
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria F'] ----

['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '*', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '*', '*', '*', '.', '#', 'Estacion U', '.']
['*', '#', '*', '*', '*', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '#', '.', '.', '.', '.']
['.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '*', '#', '=', '*', 'E', '=', '=', '#', '.']
['.', '*', 'T', '=', '#', '*', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '=', '*', '#', '=', 'Estanteria I', '.']

```

```
[ '.', '.', '.', '#', '=', '#', '=', '=', '*', '.']
[ '.', '.', '.', 'T', '=', '=', 'Estanteria K', '*', 'Estanteria L', '.']
----- El mapa resultante del algoritmo A* Tree Euclidea es -----
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1),
(1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1,
7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8,
7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5),
(1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
El coste total es de 32
La cantidad de nodos expandidos es de 1258
El tiempo total de ejecucion es: 0.34572649002075195
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria
F'] -----

['Estacion C', '*', '*', '*', '#', '#', '.', '.', '.', 'E']
['=', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['=', '=', '#', '*', '*', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['**', '**', '**', '**', '.', '#', 'Estacion U', '.']
['**', '#', '**', '**', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '*', '#', '.', '.', '.', '.', '#', '.', '.']
['*', 'E', '#', '#', '.', '#', '.', '.', '.']
['*', '=', '=', 'Estanteria F', '=', '*', '*', '#', '.', '.']
['*', '*', '#', '=', 'E', '*', '*', '#', '.']
['*', '*', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '.']
['.', '#', '**', '=', 'E', '*', '#', '=', 'Estanteria I', '.']
['.', '.', '.', '#', '=', '#', '=', '=', '*', '.']
['.', '.', '.', 'T', '=', '=', 'Estanteria K', '*', 'Estanteria L', '.']
----- El mapa resultante del algoritmo Greedy Graph Manhattan es
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3),
(0, 1, 4), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 4, 5), (0, 5, 5), (1, 5, 5), (1,
5, 6), (1, 6, 6), (1, 6, 7), (1, 7, 7), (1, 8, 7), (1, 9, 7), (1, 9, 6), (1, 9,
6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7),
(1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 35
La cantidad de nodos expandidos es de 34
El tiempo total de ejecucion es: 0.0007979869842529297
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria
F'] -----

['Estacion C', '=', '=', '=', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '=', 'T', '*', '.', '#', '.', '.']
['.', '.', '#', '=', '*', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
['.', '.', '.', '.', '=', '.', '.', '#', 'Estacion U', '.']
['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', '.', 'T', '.', '#', '.', '.', '.']
```

```
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.', '.']
[ '.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '*', '#', '=', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '=', '=', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '=', 'Estanteria L', '.']
----- El mapa resultante del algoritmo Greedy Graph Euclidea es -
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1),
(1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1,
7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8,
7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5),
(1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
El coste total es de 32
La cantidad de nodos expandidos es de 29
El tiempo total de ejecucion es: 0.0007157325744628906
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria
F'] ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '=', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '=', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '=', '=', 'Estanteria F', '=', '.', '.', '#', '.', '.']
[ '.', '.', '#', '=', 'E', '.', '.', '#', '.', '.']
[ '.', '.', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '=', '=', '*', '#', '=', 'Estanteria I', '.']
[ '.', '.', '.', '#', '=', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'T', '=', '=', 'Estanteria K', '.', 'Estanteria L', '.']
----- El mapa resultante del algoritmo Greedy Tree Manhattan es -
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3),
(0, 1, 4), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 4, 5), (0, 5, 5), (1, 5, 5), (1,
5, 6), (1, 6, 6), (1, 6, 7), (1, 7, 7), (1, 8, 7), (1, 9, 7), (1, 9, 6), (1, 9,
6), (1, 8, 6), (1, 8, 7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7),
(1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 35
La cantidad de nodos expandidos es de 38
```

```

El tiempo total de ejecucion es: 0.0008671283721923828
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria F'] ----

['Estacion C', '=', '=', '=', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '=', 'T', '*', '.', '#', '.', '.']
['.', '.', '#', '.', '=', '*', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
['.', '.', '.', '.', '=', '.', '#', 'Estacion U', '.']
['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
['.', '#', '.', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
['.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '*', '#', '=', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '=', '=', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '=', 'Estanteria L', '.']

----- El mapa resultante del algoritmo Greedy Tree Euclidea es --
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 2, 1), (0, 3, 1),
(1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1,
7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8,
7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5),
(1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
El coste total es de 32
La cantidad de nodos expandidos es de 29
El tiempo total de ejecucion es: 0.0006978511810302734
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria F'] ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '=', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '=', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '=', '=', 'Estanteria F', '=', '.', '.', '#', '.', '.']
[ '.', '.', '#', '=', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', '=', '#', '*', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '=', '#', '*', '#', '=', 'Estanteria I', '.']

```

```
[ '.', '.', '.', '#', '=', '#', '=', '=', '.', '.']
[ '.', '.', '.', 'T', '=', '=', 'Estanteria K', '.', 'Estanteria L', '.']
----- El mapa resultante del algoritmo BFS Graph es -----
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 2, 1), (0, 3, 1),
(1, 3, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 5, 3), (1, 6, 3), (1, 7, 3), (1,
7, 4), (1, 8, 4), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1, 8, 6), (1, 8,
7), (1, 7, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6),
(1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 32
La cantidad de nodos expandidos es de 228
El tiempo total de ejecucion es: 0.00873708724975586
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria
F'] -----

['Estacion C', '=', '*', '*', '#', '#', '*', '*', '*', 'E']
['*', 'Estanteria A', '#', '*', 'T', '*', '*', '#', '*', '*']
['*', '=', '#', '*', '*', 'Estanteria B', '*', 'T', '*']
['*', 'E', '#', '#', '*', '#', '*', '*', '*', '*']
['*', '*', '*', '*', '*', '*', '#', 'Estacion U', '*']
['*', '#', '*', '*', 'E', '*', '*', '#', '*']
['*', '*', 'T', '*', '#', '*', '*', '*', '*', '*']
['*', '#', '*', 'Estacion R', '*', '*', '#', '*', 'Estanteria X', '*']
['*', '*', '*', '#', '*', '*', '*', '*', '.', '.']
['*', '*', '*', 'T', '*', '*', '*', '*', '.', '.']

['*', '*', '*', '*', '*', '*', 'Estanteria J', '*', '*', 'E']
['*', 'Estanteria D', '*', 'T', '*', '*', '*', '*', '*', '*']
['*', '*', '#', '*', '*', '*', '#', '*', '*', '*']
['*', 'E', '#', '#', '*', '#', '*', '*', '*', '*']
['*', '=', '=', 'Estanteria F', '=', '=', '=', '#', '*', '*']
['*', '*', '#', '=', '*', 'E', '=', '=', '#', '*']
['*', '*', 'T', '=', '#', '*', 'G', '=', 'Estanteria H', '*']
['*', '#', '*', '=', '=', '*', '#', '=', 'Estanteria I', '*']
['*', '*', '*', '#', '=', '#', '=', '=', '*', '*']
['*', '*', '*', 'T', '=', '=', 'Estanteria K', '*', 'Estanteria L', '*']
----- El mapa resultante del algoritmo BFS Tree es -----
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: []
El coste total es de 0
La cantidad de nodos expandidos es de 0
El tiempo total de ejecucion es: 0
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria
F'] -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', '.', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estacion R', '.', '.', '#', '.', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
----- El mapa resultante del algoritmo DFS Graph es -----
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 0, 0), (0, 1, 0), (0, 2, 0), (0, 3, 0), (0, 4, 0),
(0, 5, 0), (0, 6, 0), (0, 7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0,
9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8,
9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8),
(0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 2), (0,
6, 2), (0, 6, 3), (0, 5, 3), (0, 5, 2), (0, 4, 2), (0, 4, 1), (0, 3, 1), (0, 2,
1), (0, 1, 1), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3), (0, 2, 3), (0, 2, 4),
(0, 3, 4), (0, 4, 4), (0, 5, 4), (0, 5, 5), (0, 5, 6), (0, 4, 6), (0, 3, 6), (0,
3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2, 9), (0, 2, 8), (0, 2,
7), (0, 2, 6), (0, 2, 5), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8),
(0, 1, 8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6), (1,
0, 5), (1, 0, 4), (1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0), (1, 2,
0), (1, 3, 0), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0),
(1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 6), (1,
9, 5), (1, 9, 4), (1, 9, 3), (1, 9, 2), (1, 9, 1), (1, 9, 0), (1, 8, 0), (1, 8,
1), (1, 8, 2), (1, 7, 2), (1, 7, 3), (1, 7, 4), (1, 7, 5), (1, 6, 5), (1, 6, 6),
(1, 6, 7), (1, 7, 7), (1, 8, 7), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1,
8, 8), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 5, 9), (1, 4, 9), (1, 4, 8), (1, 3,
8), (1, 3, 7), (1, 3, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 172
La cantidad de nodos expandidos es de 174
El tiempo total de ejecucion es: 0.004993915557861328
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria
F'] -----
['Estacion C', '=', '=', '=', '#', '#', '=', '=', '=', 'E']
[ '=', 'Estanteria A', '#', '=', 'T', '=', '=', '#', '=', '=']
[ '=', '=', '#', '=', '=', '=', 'Estanteria B', '=', 'T', '=']
[ '=', 'E', '#', '#', '=', '#', '=', '=', '=', '=']
[ '=', '=', '*', '=', '*', '=', '#', 'Estacion U', '=']
[ '=', '#', '=', '=', '=', 'E', '=', '*', '#', '*']
[ '=', '*', 'T', '=', '#', '=', '=', '=', '=', '=']
[ '=', '#', '=', 'Estacion R', '=', '=', '#', '=', 'Estanteria X', '=']
[ '=', '*', '*', '#', '.', '#', '*', '=', '=', '=']
[ '=', '=', 'T', '=', '=', '=', '=', '=', '=', '=']
[ '=', '=', '=', '=', '=', 'Estanteria J', '=', '=', 'E']
[ '=', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '=', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '=', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '=', '.', '.', 'Estanteria F', '=', '=', '=', '#', '=', '=']
[ '=', '.', '.', 'E', '.', '.', '#', '=', '=']
```

```

[ '=', '.', 'T', '.', '#', '=', 'G', '=', 'Estanteria H', '=']
[ '=', '#', '=', '=', '=', '#', '=', 'Estanteria I', '=']
[ '=', '=', '=', '#', '*', '#', '*', '=', '=', '=']
[ '=', '=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=']
----- El mapa resultante del algoritmo DFS Tree es -----
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: []
El coste total es de 0
La cantidad de nodos expandidos es de 0
El tiempo total de ejecucion es: 0
----- CAMINO RECORRIDO -----
----- Salida: Estacion C Destinos: ['Estanteria K', 'Estanteria H', 'Estanteria F'] ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- COMPARATIVA FINAL DE LOS ALGORITMOS -----
| Algoritmo | Tiempo Total (s) | Nodos Expandidos Totales | Coste Total |
|-----|-----|-----|
| A* Graph Manhattan | 0.005937099456787109 | 69 | 32 |
| A* Graph Euclidea | 0.005275249481201172 | 79 | 32 |
| A* Tree Manhattan | 0.0874173641204834 | 440 | 32 |
| A* Tree Euclidea | 0.34572649002075195 | 1258 | 32 |
| Greedy Graph Manhattan | 0.0007979869842529297 | 34 | 35 |
| Greedy Graph Euclidea | 0.0007157325744628906 | 29 | 32 |
| Greedy Tree Manhattan | 0.0008671283721923828 | 38 | 35 |
| Greedy Tree Euclidea | 0.0006978511810302734 | 29 | 32 |
| BFS Graph | 0.00873708724975586 | 228 | 32 |
| BFS Tree | 0 | 0 | 0 |
| DFS Graph | 0.004993915557861328 | 174 | 172 |
| DFS Tree | 0 | 0 | 0 |

```

La heurística Manhattan es más eficiente, con menos nodos expandidos (69 vs 79) y menor tiempo de ejecución , aunque ambos tienen el mismo coste total (32).

Ambas versiones de Tree son muy lentas, con A Tree Euclídea siendo especialmente lenta (1258 nodos expandidos), debido a la expansión descontrolada de nodos.

Los algoritmos Greedy Graph son los más rápidos (29-34 nodos), mientras que Greedy Tree también es rápido pero ligeramente menos eficiente.

BFS y DFS: BFS Graph es más lento que Greedy, pero más eficiente que A Tree. BFS Tree y DFS Tree no encuentran solución, en el caso de BFS porque entra en un bucle en el que vuelve a atrás y añade los mismos nodos todo el rato haciendo que la frontera pase de 1500 registros, eventualmente encontraria la solucion pero completamente insuable. DFS tree su profundidad supera el limite por lo que no tiene solucion.

Caso 2: Recogida de paquete en la planta 0: lleva el mismo al una estacion de la planta 1 , vuelve a la 0 a por otro y vuelve a subir a dejarlo en otra estanteria

```
In [134...]: analisis("Estacion U", ["Estanteria F", "Estacion R", "Estanteria J"])
```

```

Nodo Estacion U(
    Estado: Estacion U
    Coordenadas: (0, 4, 8)
    Padre: None
    Acción: [(0, 4, 8)]
    Acciones disponibles: [((0, 3, 8), 1), ((0, 4, 9), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 12
    F(n): 12
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 1
    Coste Acumulado: 14
    Heuristica: 0
    F(n): 14
    Profundidad: 10
)
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 14
La cantidad de nodos expandidos es de 15
----- CAMINO RECORRIDO -----
----- Salida: Estacion U (0, 4, 8) Destino: Estanteria F (1, 4, 3) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '.', '.', 'E', '=', '.', '#', '*']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '.', '#', '.', '.']
[ '.', '.', '#', '*', '*', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: None
)

```

```

Acción: [(1, 4, 3)]
Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
Coste: 0
Coste Acumulado: 0
Heuristica: 12
F(n): 12
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: .
    Acción: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)], Coste: 1
    Coste Acumulado: 12
    Heuristica: 0
    F(n): 12
    Profundidad: 8
)
El camino recorrido es: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
El coste total es de 12
La cantidad de nodos expandidos es de 13
----- CAMINO RECORRIDO -----
----- Salida: Estanteria F (1, 4, 3) Destino: Estacion R (0, 7, 3) -----

[['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']]
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '*', '*', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '*', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '.', '#', '.', '.']
[ '.', '.', '#', '*', '*', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: None
    Acción: [(0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)], Coste: 0
    Coste Acumulado: 0
)

```

```

Heuristica: 15
F(n): 15
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: .
    Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 1
    Coste Acumulado: 15
    Heuristica: 0
    F(n): 15
    Profundidad: 11
)
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 15
La cantidad de nodos expandidos es de 15
----- CAMINO RECORRIDO -----
----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

[['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']]
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 41
La cantidad de nodos expandidos es de 43
El tiempo total de ejecucion es: 0.003319978713989258
----- CAMINO RECORRIDO -----
----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```

```

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.', '.']
[ '.', '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '*']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '*', '*', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion U(
    Estado: Estacion U
    Coordenadas: (0, 4, 8)
    Padre: None
    Acción: [(0, 4, 8)]
    Acciones disponibles: [((0, 3, 8), 1), ((0, 4, 9), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 10.39834563766817
    F(n): 10.39834563766817
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 1
    Coste Acumulado: 14
    Heuristica: 0.0
    F(n): 14.0
    Profundidad: 10
)
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
El coste total es de 14
La cantidad de nodos expandidos es de 17
----- CAMINO RECORRIDO -----
----- Salida: Estacion U (0, 4, 8) Destino: Estanteria F (1, 4, 3) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '*', 'T', '.']

```

```
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '*' ]
[ '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*' ]
[ '.', '#', '.', '.', '.', 'E', '=', '.', '#', '*' ]
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E' ]
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '*', '.', '#', '.', '.']
[ '.', '.', '#', '*', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estanteria F(

Estado: Estanteria F
 Coordenadas: (1, 4, 3)
 Padre: None
 Acción: [(1, 4, 3)]
 Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
 Coste: 0
 Coste Acumulado: 0
 Heuristica: 10.06449510224598
 F(n): 10.06449510224598
 Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estacion R(

Estado: Estacion R
 Coordenadas: (0, 7, 3)
 Padre: .
 Acción: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
 Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
 Coste: 1
 Coste Acumulado: 12
 Heuristica: 0.0
 F(n): 12.0
 Profundidad: 8

)

El camino recorrido es: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]

El coste total es de 12

La cantidad de nodos expandidos es de 14

----- CAMINO RECORRIDO -----

----- Salida: Estanteria F (1, 4, 3) Destino: Estacion R (0, 7, 3) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '*', '*', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '*', '#', '=', '.', '.', '.', '.']
```

```
['.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '*', 'Estanteria F', '=', '=', '.', '#', '.', '.']
['.', '.', '#', '*', '*', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estacion R(

Estado: Estacion R

Coordenadas: (0, 7, 3)

Padre: None

Acción: [(0, 7, 3)]

Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 12.927446638338974

F(n): 12.927446638338974

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria J(

Estado: Estanteria J

Coordenadas: (1, 0, 6)

Padre: .

Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],

Coste: 1

Coste Acumulado: 15

Heuristica: 0.0

F(n): 15.0

Profundidad: 11

)

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 15

La cantidad de nodos expandidos es de 16

----- CAMINO RECORRIDO -----

----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
['.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
['.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
['.', '#', '*', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),
(0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1,
4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7,
3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5),
(1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 41
La cantidad de nodos expandidos es de 47
El tiempo total de ejecucion es: 0.0040400028228759766
----- CAMINO RECORRIDO -----
----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria
J'] ----

['Estacion C', '.', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '*', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '*']
[ '.', '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '*']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '*', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '*', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '*', '=', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
Nodo Estacion U(
    Estado: Estacion U
    Coordenadas: (0, 4, 8)
    Padre: None
    Acción: [(0, 4, 8)]
    Acciones disponibles: [((0, 3, 8), 1), ((0, 4, 9), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 12
    F(n): 12
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
```

```

Estado: Estanteria F
Coordenadas: (1, 4, 3)
Padre: .
Acción: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0,
5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
Coste: 1
Coste Acumulado: 14
Heuristica: 0
F(n): 14
Profundidad: 10
)
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),
(0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 14
La cantidad de nodos expandidos es de 25
----- CAMINO RECORRIDO -----
----- Salida: Estacion U (0, 4, 8) Destino: Estanteria F (1, 4, 3) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '.', '.', '.', 'E', '=', '.', '#', '*']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '.', '#', '.', '.']
[ '.', '.', '#', '*', '*', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: None
    Acción: [(1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 12
    F(n): 12
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: .
    Acción: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0,

```

```

7, 5), (0, 7, 4), (0, 7, 3)]
Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
Coste: 1
Coste Acumulado: 12
Heuristica: 0
F(n): 12
Profundidad: 8
)
El camino recorrido es: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5),
(0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
El coste total es de 12
La cantidad de nodos expandidos es de 30
----- CAMINO RECORRIDO -----
----- Salida: Estanteria F (1, 4, 3) Destino: Estacion R (0, 7, 3) -----

[['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']]
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '*', '*', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '*', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '.', '#', '.', '.']
[ '.', '.', '#', '*', '*', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: None
    Acción: [(0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 15
    F(n): 15
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: .
    Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1,
4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 1
    Coste Acumulado: 15
)

```

```

Heuristica: 0
F(n): 15
Profundidad: 11
)
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5),
(1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 15
La cantidad de nodos expandidos es de 31
----- CAMINO RECORRIDO -----
----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),
(0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1,
4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7,
3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5),
(1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 41
La cantidad de nodos expandidos es de 86
El tiempo total de ejecucion es: 0.00551605224609375
----- CAMINO RECORRIDO -----
----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria
J'] ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '.', '.']
[ '.', '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '*']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
```

```
[ '.', '.', '.', 'Estanteria F', '=', '=', '#', '.', '.']
[ '.', '.', '#', '*', '*', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion U(
    Estado: Estacion U
    Coordenadas: (0, 4, 8)
    Padre: None
    Acción: [(0, 4, 8)]
    Acciones disponibles: [((0, 3, 8), 1), ((0, 4, 9), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 10.39834563766817
    F(n): 10.39834563766817
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 1
    Coste Acumulado: 14
    Heuristica: 0.0
    F(n): 14.0
    Profundidad: 10
)
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
El coste total es de 14
La cantidad de nodos expandidos es de 33
----- CAMINO RECORRIDO -----
----- Salida: Estacion U (0, 4, 8) Destino: Estanteria F (1, 4, 3) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '*', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '*', '*']
[ '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '.', '.', '.', 'E', '=', '.', '#', '*']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '*', '.', '#', '.', '.']
[ '.', '.', '#', '*', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
```

```

[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: None
    Acción: [(1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 10.06449510224598
    F(n): 10.06449510224598
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: .
    Acción: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 1
    Coste Acumulado: 12
    Heuristica: 0.0
    F(n): 12.0
    Profundidad: 8
)
El camino recorrido es: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
El coste total es de 12
La cantidad de nodos expandidos es de 31
----- CAMINO RECORRIDO -----
----- Salida: Estanteria F (1, 4, 3) Destino: Estacion R (0, 7, 3) ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '*', '*', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '*', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '*', 'Estanteria F', '=', '=', '.', '#', '.', '.']
[ '.', '.', '#', '*', '*', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion R(
    Estado: Estacion R
)

```

```

Coordenadas: (0, 7, 3)
Padre: None
Acción: [(0, 7, 3)]
Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
Coste: 0
Coste Acumulado: 0
Heuristica: 12.927446638338974
F(n): 12.927446638338974
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: .
    Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 1
    Coste Acumulado: 15
    Heuristica: 0.0
    F(n): 15.0
    Profundidad: 11
)
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 15
La cantidad de nodos expandidos es de 35
----- CAMINO RECORRIDO -----
----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
[ '.', '#', '*', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
```

El coste total es de 41
 La cantidad de nodos expandidos es de 99
 El tiempo total de ejecucion es: 0.004470348358154297
 ----- CAMINO RECORRIDO -----
 ----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '*', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '*']
[ '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '*']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '*', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '*', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '*', '=', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion U(
  Estado: Estacion U
  Coordenadas: (0, 4, 8)
  Padre: None
  Acción: [(0, 4, 8)]
  Acciones disponibles: [((0, 3, 8), 1), ((0, 4, 9), 1)],
  Coste: 0
  Coste Acumulado: 0
  Heuristica: 12
  F(n): 12
  Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
  Estado: Estanteria F
  Coordenadas: (1, 4, 3)
  Padre: .
  Acción: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
  Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
  Coste: 1
  Coste Acumulado: 14
  Heuristica: 0
  F(n): 0
  Profundidad: 10
)
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 14
La cantidad de nodos expandidos es de 10
```

----- CAMINO RECORRIDO -----
 ----- Salida: Estacion U (0, 4, 8) Destino: Estanteria F (1, 4, 3) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estanteria F(

```
Estado: Estanteria F
Coordenadas: (1, 4, 3)
Padre: None
Acción: [(1, 4, 3)]
Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
Coste: 0
Coste Acumulado: 0
Heuristica: 12
F(n): 12
Profundidad: 0
```

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estacion R(

```
Estado: Estacion R
Coordenadas: (0, 7, 3)
Padre: .
Acción: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
Coste: 1
Coste Acumulado: 12
Heuristica: 0
F(n): 0
Profundidad: 8
```

)

El camino recorrido es: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]

El coste total es de 12

La cantidad de nodos expandidos es de 8

----- CAMINO RECORRIDO -----

----- Salida: Estanteria F (1, 4, 3) Destino: Estacion R (0, 7, 3) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
```

```
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estacion R(

Estado: Estacion R
 Coordenadas: (0, 7, 3)
 Padre: None
 Acción: [(0, 7, 3)]
 Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
 Coste: 0
 Coste Acumulado: 0
 Heuristica: 15
 F(n): 15
 Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria J(

Estado: Estanteria J
 Coordenadas: (1, 0, 6)
 Padre: .
 Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
 Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
 Coste: 1
 Coste Acumulado: 15
 Heuristica: 0
 F(n): 0
 Profundidad: 11

)

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 15

La cantidad de nodos expandidos es de 11

----- CAMINO RECORRIDO -----

----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
```

```
['.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
['.', '.', 'T', '=', '#', '.', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
['.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 29

El tiempo total de ejecucion es: 0.0007808208465576172

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
['.', '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
['.', '#', '.', '=', '=', 'E', '=', '.', '#', '.']
['.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estacion U(

Estado: Estacion U

Coordenadas: (0, 4, 8)

Padre: None

Acción: [(0, 4, 8)]

Acciones disponibles: [(0, 3, 8), (0, 4, 9)],

Coste: 0

Coste Acumulado: 0

Heuristica: 10.39834563766817

F(n): 10.39834563766817

```

Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 1
    Coste Acumulado: 14
    Heuristica: 0.0
    F(n): 0.0
    Profundidad: 10
)
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
El coste total es de 14
La cantidad de nodos expandidos es de 10
----- CAMINO RECORRIDO -----
----- Salida: Estacion U (0, 4, 8) Destino: Estanteria F (1, 4, 3) -----

[['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']],
[('.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']],
[('.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.')],
[('.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']],
[('.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.')],
[('.', '#', '.', '.', '.', 'E', '=', '.', '#', '.']],
[('.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']],
[('.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.')],
[('.', '.', '.', '#', '.', '#', '.', '.', '.', '.']],
[('.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']]

[('.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E')],
[('.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']],
[('.', '.', '#', '.', '.', '.', '.', '#', '.', '.']],
[('.', 'E', '#', '#', '.', '#', '.', '.', '.']),
[('.', '.', '.', 'Estanteria F', '=', '.', '.', '#', '.', '.']),
[('.', '.', '#', '.', '=', 'E', '.', '.', '#', '.']),
[('.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.')],
[('.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.')],
[('.', '.', '.', '#', '.', '#', '.', '.', '.', '.']],
[('.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.')]

Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: None
    Acción: [(1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 10.06449510224598
    F(n): 10.06449510224598
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----

```

El nodo resultante es:

```
Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: .
    Acción: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 1
    Coste Acumulado: 12
    Heuristica: 0.0
    F(n): 0.0
    Profundidad: 8
)
```

El camino recorrido es: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]

El coste total es de 12

La cantidad de nodos expandidos es de 8

----- CAMINO RECORRIDO -----

----- Salida: Estanteria F (1, 4, 3) Destino: Estacion R (0, 7, 3) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estacion R(

```
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: None
    Acción: [(0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 12.927446638338974
    F(n): 12.927446638338974
    Profundidad: 0
)
```

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

```
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
```

Padre: .
 Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
 Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
 Coste: 1
 Coste Acumulado: 15
 Heuristica: 0.0
 F(n): 0.0
 Profundidad: 11
)
 El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
 El coste total es de 15
 La cantidad de nodos expandidos es de 11
 ----- CAMINO RECORRIDO -----
 ----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

```
[['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']]
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----  

    El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]  

    El coste total es de 41  

    La cantidad de nodos expandidos es de 29  

    El tiempo total de ejecucion es: 0.0012278556823730469  

    ----- CAMINO RECORRIDO -----  

    ----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----  


```
[['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']]
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '=', '.', '=', '.']
['.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
['.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
['.', '.', 'T', '=', '#', '.', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
```


```

```
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estacion U(

Estado: Estacion U

Coordenadas: (0, 4, 8)

Padre: None

Acción: [(0, 4, 8)]

Acciones disponibles: [((0, 3, 8), 1), ((0, 4, 9), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 12

F(n): 12

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria F(

Estado: Estanteria F

Coordenadas: (1, 4, 3)

Padre: .

Acción: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3)]

Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],

Coste: 1

Coste Acumulado: 14

Heuristica: 0

F(n): 0

Profundidad: 10

)

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3)]

El coste total es de 14

La cantidad de nodos expandidos es de 11

----- CAMINO RECORRIDO -----

----- Salida: Estacion U (0, 4, 8) Destino: Estanteria F (1, 4, 3) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '.', '.']
[ '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
[ '.', '.', '#', '.', '.', '.', '#', '.', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estanteria F(

Estado: Estanteria F

Coordenadas: (1, 4, 3)

Padre: None

Acción: [(1, 4, 3)]

Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 12

F(n): 12

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estacion R(

Estado: Estacion R

Coordenadas: (0, 7, 3)

Padre: .

Acción: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]

Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],

Coste: 1

Coste Acumulado: 12

Heuristica: 0

F(n): 0

Profundidad: 8

)

El camino recorrido es: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]

El coste total es de 12

La cantidad de nodos expandidos es de 8

----- CAMINO RECORRIDO -----

----- Salida: Estanteria F (1, 4, 3) Destino: Estacion R (0, 7, 3) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '#', '.', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
```

```

[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: None
    Acción: [(0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 15
    F(n): 15
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: .
    Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 1
    Coste Acumulado: 15
    Heuristica: 0
    F(n): 0
    Profundidad: 11
)
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 15
La cantidad de nodos expandidos es de 11
----- CAMINO RECORRIDO -----
----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

```

----- RESULTADO FINAL DE LA BUSQUEDA -----
 El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),
 (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1,
 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7,
 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5),
 (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
 El coste total es de 41
 La cantidad de nodos expandidos es de 30
 El tiempo total de ejecucion es: 0.0020513534545898438
 ----- CAMINO RECORRIDO -----
 ----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion U(
  Estado: Estacion U
  Coordenadas: (0, 4, 8)
  Padre: None
  Acción: [(0, 4, 8)]
  Acciones disponibles: [((0, 3, 8), 1), ((0, 4, 9), 1)],
  Coste: 0
  Coste Acumulado: 0
  Heuristica: 10.39834563766817
  F(n): 10.39834563766817
  Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
  Estado: Estanteria F
  Coordenadas: (1, 4, 3)
  Padre: .
  Acción: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0,
  5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
  Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
  Coste: 1
  Coste Acumulado: 14
  Heuristica: 0.0
  F(n): 0.0

```

```

Profundidad: 10
)
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),
(0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3)]
El coste total es de 14
La cantidad de nodos expandidos es de 10
----- CAMINO RECORRIDO -----
----- Salida: Estacion U (0, 4, 8) Destino: Estanteria F (1, 4, 3) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: None
    Acción: [(1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 10.06449510224598
    F(n): 10.06449510224598
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: .
    Acción: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0,
    7, 5), (0, 7, 4), (0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 1
    Coste Acumulado: 12
    Heuristica: 0.0
    F(n): 0.0
    Profundidad: 8
)
El camino recorrido es: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5),
(0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
```

El coste total es de 12
 La cantidad de nodos expandidos es de 8
 ----- CAMINO RECORRIDO -----
 ----- Salida: Estanteria F (1, 4, 3) Destino: Estacion R (0, 7, 3) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion R(
  Estado: Estacion R
  Coordenadas: (0, 7, 3)
  Padre: None
  Acción: [(0, 7, 3)]
  Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
  Coste: 0
  Coste Acumulado: 0
  Heuristica: 12.927446638338974
  F(n): 12.927446638338974
  Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria J(
  Estado: Estanteria J
  Coordenadas: (1, 0, 6)
  Padre: .
  Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
  Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
  Coste: 1
  Coste Acumulado: 15
  Heuristica: 0.0
  F(n): 0.0
  Profundidad: 11
)
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 15
La cantidad de nodos expandidos es de 11
----- CAMINO RECORRIDO -----
----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----
```

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----

```

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 29

El tiempo total de ejecucion es: 0.0020072460174560547

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '=', '.']
[ '.', '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- NODO -----

```

Nodo Estacion U(

Estado: Estacion U

Coordenadas: (0, 4, 8)

Padre: None

```

Acción: [(0, 4, 8)]
Acciones disponibles: [((0, 3, 8), 1), ((0, 4, 9), 1)],
Coste: 0
Coste Acumulado: 0
Heuristica: 12
F(n): 12
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: .
    Acción: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)], Coste: 1
    Coste Acumulado: 14
    Heuristica: 0
    F(n): 0
    Profundidad: 10
)
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3)]
El coste total es de 14
La cantidad de nodos expandidos es de 87
----- CAMINO RECORRIDO -----
----- Salida: Estacion U (0, 4, 8) Destino: Estanteria F (1, 4, 3) -----

['Estacion C', '.', '*', '*', '#', '#', '*', '*', '*', 'E']
[ '.', 'Estanteria A', '#', '*', 'T', '*', '*', '#', '*', '*']
[ '.', '.', '#', '*', '*', '*', 'Estanteria B', '*', 'T', '*']
[ '.', 'E', '#', '#', '*', '#', '=', '=', '=', '*']
[ '.', '*', '*', '*', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '*', '*', 'E', '=', '*', '#', '*']
[ '.', '.', 'T', '*', '#', '*', '*', '*', '*', '*']
[ '.', '#', '.', 'Estacion R', '*', '*', '#', '*', 'Estanteria X', '*']
[ '.', '.', '.', '#', '*', '#', '*', '*', '*']
[ '.', '.', '.', 'T', '.', '.', '*', '*', '*', '*']

[ '.', '.', '.', '.', '.', '*', 'Estanteria J', '*', '*', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '*', '*', '*', '*']
[ '.', '.', '#', '.', '.', '.', '#', '*', '*', '*']
[ '.', 'E', '#', '#', '*', '#', '*', '.', '*', '*']
[ '.', '.', '.', 'Estanteria F', '=', '=', '*', '#', '.', '*']
[ '.', '.', '#', '*', '*', 'E', '*', '*', '#', '.']
[ '.', '.', 'T', '.', '#', '*', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '*', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria F(
    Estado: Estanteria F
    Coordenadas: (1, 4, 3)
    Padre: None
    Acción: [(1, 4, 3)]
    Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)], Coste: 0
    Coste Acumulado: 0
)

```

```

Heuristica: 12
F(n): 12
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: .
    Acción: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 1
    Coste Acumulado: 12
    Heuristica: 0
    F(n): 0
    Profundidad: 8
)
El camino recorrido es: [(1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
El coste total es de 12
La cantidad de nodos expandidos es de 109
----- CAMINO RECORRIDO -----
----- Salida: Estanteria F (1, 4, 3) Destino: Estacion R (0, 7, 3) -----

[['Estacion C', '*', '.', '.', '#', '#', '.', '.', '.', 'E']]
['*', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[['*', '*', '#', '.', '*', '.', 'Estanteria B', '.', 'T', '.']]
[['*', 'E', '#', '#', '*', '#', '*', '*', '.', '.']]
[['*', '*', '*', '*', '*', '*', '#', 'Estacion U', '.']]
[['*', '#', '*', '*', '*', 'E', '*', '*', '#', '.']]
[['.', '.', 'T', '*', '#', '=', '*', '*', '*', '.']]
[['.', '#', '.', 'Estacion R', '=', '=', '#', '*', 'Estanteria X', '.']]
[['.', '.', '.', '#', '*', '#', '.', '.', '.', '.']]
[['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']]

[['*', '*', '*', '*', '*', 'Estanteria J', '*', '.', 'E']]
[['*', 'Estanteria D', '*', 'T', '*', '*', '*', '*', '*', '.']]
[['*', '#', '*', '*', '*', '#', '*', '*', '*']]
[['*', 'E', '#', '#', '#', '*', '*', '*', '*']]
[['*', '*', '*', 'Estanteria F', '=', '=', '*', '#', '*', '*']]
[['*', '#', '*', '*', 'E', '*', '*', '#', '.']]
[['*', '*', 'T', '*', '#', '*', 'G', '*', 'Estanteria H', '*']]
[['*', '#', '*', '*', '*', '#', '*', 'Estanteria I', '.']]
[['*', '*', '#', '*', '#', '.', '*', '.', '.']]
[['.', '*', '*', 'T', '*', '*', 'Estanteria K', '.', 'Estanteria L', '.']]

Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: None
    Acción: [(0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 15
    F(n): 15
    Profundidad: 0
)

```

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria J(

Estado: Estanteria J

Coordenadas: (1, 0, 6)

Padre: .

Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],

Coste: 1

Coste Acumulado: 15

Heuristica: 0

F(n): 0

Profundidad: 11

)

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 15

La cantidad de nodos expandidos es de 134

----- CAMINO RECORRIDO -----

----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

```
['Estacion C', '*', '*', '*', '#', '#', '*', '*', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '*', '*', '#', '*', '.']
['*', '*', '#', '*', '*', 'Estanteria B', '*', 'T', '*']
['*', 'E', '#', '#', '#', '*', '*', '*', '*']
['*', '*', '*', '*', '*', '*', '#', 'Estacion U', '*']
['*', '#', '*', '=', '=', 'E', '*', '*', '#', '*']
['*', '*', 'T', '=', '#', '*', '*', '*', '*']
['*', '#', '*', 'Estacion R', '*', '*', '#', '*', 'Estanteria X', '*']
['*', '*', '#', '*', '#', '*', '*', '*', '*']
['*', '*', '*', 'T', '*', '*', '*', '*', '*']

['*', '*', '*', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['*', 'Estanteria D', '*', 'T', '*', '.', '=', '.', '.', '.']
['*', '*', '#', '*', '*', '=', '#', '.', '.']
['*', 'E', '#', '#', '#', '=', '*', '*', '.']
['*', '*', '*', 'Estanteria F', '*', '=', '=', '#', '.', '.']
['*', '*', '#', '*', 'E', '*', '*', '#', '.']
['*', '*', 'T', '*', '#', '*', 'G', '*', 'Estanteria H', '*']
['.', '#', '*', '*', '#', '#', '*', 'Estanteria I', '.']
['.', '.', '.', '#', '*', '#', '.', '*', '.', '.']
['.', '.', '.', 'T', '*', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 330

El tiempo total de ejecucion es: 0.011821508407592773

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '*', '*', '*', '#', '#', '*', '*', '*', 'E']
['*', 'Estanteria A', '#', '*', 'T', '*', '*', '#', '*', '*']
['*', '*', '#', '*', '*', 'Estanteria B', '*', 'T', '*']
```

```
[ '*', 'E', '#', '#', '*', '#', '=', '=', '=', '=', '*' ]
[ '*', '*', '*', '*', '*', '*', '=', '#', 'Estacion U', '*' ]
[ '*', '#', '*', '=', '=', 'E', '=', '*', '#', '*' ]
[ '*', '*', 'T', '=', '#', '=', '*', '*', '*', '*' ]
[ '*', '#', '*', 'Estacion R', '=', '=', '#', '*', 'Estanteria X', '*' ]
[ '*', '*', '#', '*', '#', '*', '*', '*', '*' ]
[ '*', '*', '*', 'T', '*', '*', '*', '*', '*', '*' ]
```

```
[ '*', '*', '*', '*', '*', 'Estanteria J', '*', '*', 'E' ]
[ '*', 'Estanteria D', '*', 'T', '*', '*', '=', '*', '*', '*' ]
[ '*', '*', '#', '*', '*', '*', '=', '#', '*', '*' ]
[ '*', 'E', '#', '#', '*', '#', '=', '*', '*', '*' ]
[ '*', '*', '*', 'Estanteria F', '=', '=', '=', '#', '*', '*' ]
[ '*', '*', '#', '*', '*', 'E', '*', '*', '#', '.' ]
[ '*', '*', 'T', '*', '#', '*', 'G', '*', 'Estanteria H', '*' ]
[ '*', '#', '*', '*', '*', '#', '*', 'Estanteria I', '.' ]
[ '*', '*', '#', '*', '#', '.', '*', '.', '.' ]
[ '.', '*', '*', 'T', '*', '*', 'Estanteria K', '.', 'Estanteria L', '.' ]
```

Nodo Estacion U(

Estado: Estacion U

Coordenadas: (0, 4, 8)

Padre: None

Acción: [(0, 4, 8)]

Acciones disponibles: [((0, 3, 8), 1), ((0, 4, 9), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 12

F(n): 12

Profundidad: 0

)

EL RESULTADO NO PUEDE SER ENCONTRADO

EL PROBLEMA NO TIENE SOLUCION: No se ha encontrado ruta desde Estacion U a Estanteria F

Nodo Estacion U(

Estado: Estacion U

Coordenadas: (0, 4, 8)

Padre: None

Acción: [(0, 4, 8)]

Acciones disponibles: [((0, 3, 8), 1), ((0, 4, 9), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 12

F(n): 12

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria F(

Estado: Estanteria F

Coordenadas: (1, 4, 3)

Padre: .

Acción: [(0, 4, 8), (0, 4, 9), (0, 5, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8, 1), (0, 8, 0), (0, 7, 0), (0, 6, 0), (0, 6, 1), (0, 6, 2), (0, 6, 3), (0, 5, 3), (0, 5, 2), (0, 4, 2), (0, 4, 1), (0, 4, 0), (0, 3, 0), (0, 3, 1), (0, 2, 1), (0, 2, 0), (0, 1, 0), (0, 1, 1), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3), (0, 2, 3), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 5, 4), (0, 5, 5), (0, 5, 6), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 3, 9), (0, 2, 9), (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1,

```

8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6), (1, 0, 5),
(1, 0, 4), (1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0), (1, 2, 0), (1,
3, 0), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9,
1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8),
(1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1,
6, 9), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7,
3), (1, 7, 2), (1, 6, 2), (1, 6, 1), (1, 5, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3)]
Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
Coste: 1
Coste Acumulado: 116
Heuristica: 0
F(n): 0
Profundidad: 108
)

```

El camino recorrido es: [(0, 4, 8), (0, 4, 9), (0, 5, 9), (0, 6, 9), (0, 6, 8),
(0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 2), (0,
8, 2), (0, 8, 1), (0, 8, 0), (0, 7, 0), (0, 6, 0), (0, 6, 1), (0, 6, 2), (0, 6,
3), (0, 5, 3), (0, 5, 2), (0, 4, 2), (0, 4, 1), (0, 4, 0), (0, 3, 0), (0, 3, 1),
(0, 2, 1), (0, 2, 0), (0, 1, 0), (0, 1, 1), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0,
1, 3), (0, 2, 3), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 5, 4), (0, 5, 5), (0, 5,
6), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 3, 9), (0, 2, 9), (0, 2, 8),
(0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0,
0, 8), (0, 1, 8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0,
6), (1, 0, 5), (1, 0, 4), (1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0),
(1, 2, 0), (1, 3, 0), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1,
9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9,
7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8),
(1, 7, 9), (1, 6, 9), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1,
7, 4), (1, 7, 3), (1, 7, 2), (1, 6, 2), (1, 6, 1), (1, 5, 1), (1, 4, 1), (1, 4,
2), (1, 4, 3)]

El coste total es de 116

La cantidad de nodos expandidos es de 135

----- CAMINO RECORRIDO -----

----- Salida: Estacion U (0, 4, 8) Destino: Estanteria F (1, 4, 3) -----

```

['Estacion C', '=', '=', '=', '#', '#', '=', '=', '=', 'E']
['=', 'Estanteria A', '#', '=', 'T', '=', '=', '#', '=', '=']
['=', '=', '#', '=', '=', '=', 'Estanteria B', '=', 'T', '=']
['=', 'E', '#', '#', '=', '#', '=', '=', '=', '=']
['=', '=', '*', '=', '*', '=', '#', 'Estacion U', '=']
['*', '#', '=', '=', 'E', '=', '*', '#', '=']
['=', '=', 'T', '=', '#', '=', '=', '=', '=', '=']
['=', '#', '=', 'Estacion R', '=', '=', '#', '*', 'Estanteria X', '*']
['=', '=', '#', '*', '#', '*', '*', '*', '*']
['*', '*', '*', 'T', '*', '*', '*', '*', '*', '*']

```

```

['=', '=', '=', '=', '=', 'Estanteria J', '=', '=', 'E']
['=', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['=', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['=', 'E', '#', '#', '#', '.', '.', '.', '.', '.']
['=', '=', '=', 'Estanteria F', '.', '.', '.', '#', '.', '.']
['=', '=', '#', '.', '.', 'E', '.', '.', '#', '.']
['=', '=', 'T', '.', '#', '=', 'G', '=', 'Estanteria H', '=']
['=', '#', '=', '=', '=', '#', '=', 'Estanteria I', '=']
['=', '*', '*', '#', '.', '#', '*', '=', '=', '=']
['=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=']

```

Nodo Estanteria F(

Estado: Estanteria F

Coordenadas: (1, 4, 3)

Padre: None

```

Acción: [(1, 4, 3)]
Acciones disponibles: [((1, 4, 4), 1), ((1, 5, 3), 1), ((1, 4, 2), 1)],
Coste: 0
Coste Acumulado: 0
Heuristica: 12
F(n): 12
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: .
        Acción: [(1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7, 3), (1, 7, 2), (1, 6, 2), (1, 6, 3), (1, 5, 3), (1, 5, 4), (1, 5, 5), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 3, 4), (1, 2, 4), (1, 2, 3), (1, 1, 3), (1, 1, 2), (1, 1, 1), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 3, 1), (0, 3, 1), (0, 3, 0), (0, 4, 0), (0, 5, 0), (0, 6, 0), (0, 7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
        Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
        Coste: 1
        Coste Acumulado: 90
        Heuristica: 0
        F(n): 0
        Profundidad: 82
)
El camino recorrido es: [(1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7, 3), (1, 7, 2), (1, 6, 2), (1, 6, 3), (1, 5, 3), (1, 5, 4), (1, 5, 5), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 3, 4), (1, 2, 4), (1, 2, 3), (1, 1, 3), (1, 1, 2), (1, 1, 1), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 3, 1), (0, 3, 1), (0, 3, 0), (0, 4, 0), (0, 5, 0), (0, 6, 0), (0, 7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3)]
El coste total es de 90
La cantidad de nodos expandidos es de 90
----- CAMINO RECORRIDO -----
----- Salida: Estanteria F (1, 4, 3) Destino: Estacion R (0, 7, 3) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '=', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '=', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '=', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '=', '.', 'T', '.', '#', '=', '=', '=', '=', '=']
[ '=', '#', '.', 'Estacion R', '=', '=', '#', '=', '=', 'Estanteria X', '=']
[ '=', '.', '.', '#', '.', '#', '*', '=', '=', '=']

```

```
[ '=', '=', '=', 'T', '=', '=', '=', '=', '=', '=']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', '.', 'E']
['=', 'Estanteria D', '=', 'T', '.', '.', '.', '.', '.', '.']
['*', '#', '=', '=', '.', '.', '#', '.', '.']
['E', '#', '#', '=', '#', '.', '.', '.', '.']
['=', '=', '=', 'Estanteria F', '=', '=', '=', '=', '#', '.', '.']
['*', '#', '=', '=', 'E', '=', '*', '#', '.']
['*', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '=']
['#', '=', '=', '=', '=', '#', '=', 'Estanteria I', '=']
['*', '#', '.', '#', '*', '=', '=', '=']
['=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=']
```

Nodo Estacion R(

Estado: Estacion R

Coordenadas: (0, 7, 3)

Padre: None

Acción: [(0, 7, 3)]

Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 15

F(n): 15

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria J(

Estado: Estanteria J

Coordenadas: (1, 0, 6)

Padre: .

```
Acción: [(0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8, 1), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 5, 5), (0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 5, 0), (0, 4, 0), (0, 4, 1), (0, 4, 2), (0, 4, 3), (0, 4, 4), (0, 4, 5), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2, 9), (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 1, 4), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6)]
```

Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],

Coste: 1

Coste Acumulado: 75

Heuristica: 0

F(n): 0

Profundidad: 67

)

El camino recorrido es: [(0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8, 1), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 5, 5), (0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 5, 0), (0, 4, 0), (0, 4, 1), (0, 4, 2), (0, 4, 3), (0, 4, 4), (0, 4, 5), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2, 9), (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 1, 4), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6)]

El coste total es de 75

La cantidad de nodos expandidos es de 75

----- CAMINO RECORRIDO -----

----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

```
['Estacion C', '.', '.', '.', '#', '#', '=', '=', '=', 'E']
[ '.', 'Estanteria A', '#', '=', 'T', '=', '#', '=', '#', '=']
[ '.', '.', '#', '=', '=', 'Estanteria B', '=', 'T', '=']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '=']
[ '=', '=', '=', '=', '=', '#', 'Estacion U', '=']
[ '=', '#', '=', '=', '=', 'E', '*', '*', '#', '*']
[ '=', '=', 'T', '.', '#', '=', '=', '=', '=', '=']
[ '*', '#', '=', 'Estacion R', '*', '*', '#', '=', 'Estanteria X', '=']
[ '=', '=', '#', '*', '#', '*', '=', '=', '=', '=']
[ '=', '=', 'T', '=', '=', '=', '=', '=', '=', '=']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '=', '=', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 4, 9), (0, 5, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8, 1), (0, 8, 0), (0, 7, 0), (0, 6, 0), (0, 6, 1), (0, 6, 2), (0, 6, 3), (0, 5, 3), (0, 5, 2), (0, 4, 2), (0, 4, 1), (0, 4, 0), (0, 3, 0), (0, 3, 1), (0, 2, 1), (0, 2, 0), (0, 1, 0), (0, 1, 1), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3), (0, 2, 3), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 5, 4), (0, 5, 5), (0, 5, 6), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 3, 9), (0, 2, 9), (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6), (1, 0, 5), (1, 0, 4), (1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7, 3), (1, 7, 2), (1, 6, 2), (1, 6, 3), (1, 5, 3), (1, 5, 4), (1, 5, 5), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 3, 4), (1, 2, 4), (1, 2, 3), (1, 1, 3), (1, 1, 2), (1, 1, 1), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 3, 1), (0, 3, 1), (0, 3, 0), (0, 4, 0), (0, 5, 0), (0, 6, 0), (0, 7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8, 1), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8, 1), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 5, 5), (0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 5, 0), (0, 4, 0), (0, 4, 1), (0, 4, 2), (0, 4, 3), (0, 4, 4), (0, 4, 5), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2, 9), (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 2, 4), (0, 2, 3), (0, 2, 2), (0, 2, 1), (0, 2, 0), (0, 1, 9), (0, 1, 8), (0, 1, 7), (0, 1, 6), (0, 1, 5), (0, 1, 4), (0, 1, 3), (0, 1, 2), (0, 1, 1), (0, 1, 0), (0, 0, 9), (0, 0, 8), (0, 0, 7), (0, 0, 6), (0, 0, 5), (0, 0, 4), (0, 0, 3), (0, 0, 2), (0, 0, 1), (0, 0, 0)]

```

1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6)]
El coste total es de 281
La cantidad de nodos expandidos es de 300
El tiempo total de ejecucion es: 0.008786439895629883
----- CAMINO RECORRIDO -----
----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] ----

['Estacion C', '=', '=', '=', '#', '#', '=', '=', '=', 'E']
[ '=', 'Estanteria A', '#', '=', 'T', '=', '=', '#', '=', '=']
[ '=', '=', '#', '=', '=', '=', 'Estanteria B', '=', 'T', '=']
[ '=', 'E', '#', '#', '=', '#', '=', '=', '=', '=']
[ '=', '=', '=', '=', '=', '=', '#', 'Estacion U', '=']
[ '=', '#', '=', '=', '=', 'E', '=', '*', '#', '=']
[ '=', '=', 'T', '=', '#', '=', '=', '=', '=', '=']
[ '=', '#', '=', 'Estacion R', '=', '=', '#', '=', 'Estanteria X', '=']
[ '=', '=', '#', '*', '#', '*', '=', '=', '=']
[ '=', '=', 'T', '=', '=', '=', '=', '=', '=', '=']

[ '=', '=', '=', '=', '=', 'Estanteria J', '=', '=', 'E']
[ '=', 'Estanteria D', '=', 'T', '.', '.', '.', '.', '.', '.']
[ '=', '*', '#', '=', '=', '.', '.', '#', '.', '.']
[ '=', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
[ '=', '=', '=', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '=', '=', '#', '=', 'E', '=', '*', '#', '.']
[ '=', '=', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '=']
[ '=', '#', '=', '=', '=', '#', '=', 'Estanteria I', '=']
[ '=', '*', '*', '#', '.', '#', '*', '=', '=', '=']
[ '=', '=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=']
Nodo Estacion U(
    Estado: Estacion U
    Coordenadas: (0, 4, 8)
    Padre: None
    Accion: [(0, 4, 8)]
    Acciones disponibles: [((0, 3, 8), 1), ((0, 4, 9), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 12
    F(n): 12
    Profundidad: 0
)
EL RESULTADO NO PUEDE SER ENCONTRADO
EL PROBLEMA NO TIENE SOLUCION: No se ha encontrado ruta desde Estacion U a Estanteria F
----- El mapa resultante del algoritmo A* Graph Manhattan es -----
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),
(0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1,
4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7,
3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5),
(1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 41
La cantidad de nodos expandidos es de 43
El tiempo total de ejecucion es: 0.003319978713989258
----- CAMINO RECORRIDO -----
----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']

```

```
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '.', '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '*']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=' , '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '*', '*', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
----- El mapa resultante del algoritmo A* Graph Euclidea es -----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),
(0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1,
4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7,
3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5),
(1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 41
La cantidad de nodos expandidos es de 47
El tiempo total de ejecucion es: 0.0040400028228759766
----- CAMINO RECORRIDO -----
----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria
J'] ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '*', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '.', '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '*']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '*', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=' , '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '*', '*', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
----- El mapa resultante del algoritmo A* Tree Manhattan es -----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),
```

```
(0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
```

El coste total es de 41

La cantidad de nodos expandidos es de 86

El tiempo total de ejecucion es: 0.00551605224609375

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
['.', '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
['.', '#', '.', '=', '=', 'E', '=', '.', '#', '*']
['.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
['.', '#', '.', '=', '#', '.', '.', '.', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '.', '#', '*', '*', 'E', '*', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

----- El mapa resultante del algoritmo A* Tree Euclidea es -----

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 99

El tiempo total de ejecucion es: 0.004470348358154297

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', '.', 'Estanteria B', '*', 'T', '.']
['.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
['.', '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
['.', '#', '.', '=', '=', 'E', '=', '.', '#', '*']
['.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
['.', '#', '*', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
```

```
['.', '.', '#', '.', '.', '=', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
['.', '.', '*', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '.', '#', '*', '=', 'E', '*', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo Greedy Graph Manhattan es -----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 29

El tiempo total de ejecucion es: 0.0007808208465576172

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
['.', '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
['.', '#', '.', '=', '=', 'E', '=', '.', '#', '.']
['.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

-----
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo Greedy Graph Euclidea es -----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 29

El tiempo total de ejecucion es: 0.012278556823730469

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
```

```
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '.', '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '...', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=' , '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo Greedy Tree Manhattan es -
-----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 30

El tiempo total de ejecucion es: 0.0020513534545898438

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
[ 'Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '.', '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '...', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=' , '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo Greedy Tree Euclidea es -
-----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),

```
(0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
```

El coste total es de 41

La cantidad de nodos expandidos es de 29

El tiempo total de ejecucion es: 0.0020072460174560547

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
['.', '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
['.', '#', '.', '=', '=', 'E', '=', '.', '#', '.']
['.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '.', '#', '.', '=', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

----- El mapa resultante del algoritmo BFS Graph es -----

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 330

El tiempo total de ejecucion es: 0.011821508407592773

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '*', '*', '*', '#', '#', '*', '*', '*', 'E']
['*', 'Estanteria A', '#', '*', 'T', '*', '*', '#', '*', '*']
['*', '*', '#', '*', '*', '*', 'Estanteria B', '*', 'T', '*']
['*', 'E', '#', '#', '#', '=', '=', '=', '*']
['*', '*', '*', '*', '*', '=', '#', 'Estacion U', '*']
['*', '#', '*', '=', '=', 'E', '=', '*', '#', '*']
['*', '*', 'T', '=', '#', '=', '*', '*', '*', '*']
['*', '#', '*', 'Estacion R', '=', '=', '#', '*', '*', '*']
['*', '*', '#', '*', '#', '*', '*', '*', '*', '*']
['*', '*', '*', 'T', '*', '*', '*', '*', '*', '*', '*']

['*', '*', '*', '*', '*', '*', 'Estanteria J', '*', '*', 'E']
['*', 'Estanteria D', '*', 'T', '*', '*', '=', '*', '*', '*', '*']
```

```
[ '*', '*', '#', '*', '*', '=', '#', '*', '*' ]
[ '*', 'E', '#', '#', '*', '#', '=', '*', '*', '*' ]
[ '*', '*', '*', 'Estanteria F', '=', '=', '=', '#', '*', '*' ]
[ '*', '*', '#', '*', '*', 'E', '*', '*', '#', '.', '']
[ '*', '*', 'T', '*', '#', '*', 'G', '*', 'Estanteria H', '*' ]
[ '#', '*', '*', '*', '*', '#', '*', 'Estanteria I', '.' ]
[ '*', '*', '*', '#', '*', '#', '.', '*', '.', '.', '']
[ '.', '*', '*', 'T', '*', '*', 'Estanteria K', '.', 'Estanteria L', '.']
----- El mapa resultante del algoritmo BFS Tree es -----
-----
```

```
----- RESULTADO FINAL DE LA BUSQUEDA -----
```

El camino recorrido es: []

El coste total es de 0

La cantidad de nodos expandidos es de 0

El tiempo total de ejecucion es: 0

```
----- CAMINO RECORRIDO -----
```

```
----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----
```

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

-----
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

----- El mapa resultante del algoritmo DFS Graph es -----
-----
```

```
----- RESULTADO FINAL DE LA BUSQUEDA -----
```

El camino recorrido es: [(0, 4, 8), (0, 4, 9), (0, 5, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8, 1), (0, 8, 0), (0, 7, 0), (0, 6, 0), (0, 6, 1), (0, 6, 2), (0, 6, 3), (0, 5, 3), (0, 5, 2), (0, 4, 2), (0, 4, 1), (0, 4, 0), (0, 3, 0), (0, 3, 1), (0, 2, 1), (0, 2, 0), (0, 1, 0), (0, 1, 1), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0, 1, 3), (0, 2, 3), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 5, 4), (0, 5, 5), (0, 5, 6), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 3, 9), (0, 2, 9), (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6), (1, 0, 5), (1, 0, 4), (1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7, 3), (1, 7, 2), (1, 6, 2), (1, 6, 1), (1, 5, 1), (1, 4, 1), (1, 4, 2), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1,

```

9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8,
7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 6, 8), (1, 6, 7), (1, 6, 6),
(1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7, 3), (1, 7, 2), (1, 6, 2), (1, 6, 3), (1,
5, 3), (1, 5, 4), (1, 5, 5), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 3,
4), (1, 2, 4), (1, 2, 3), (1, 1, 3), (1, 1, 2), (1, 1, 1), (1, 1, 0), (1, 2, 0),
(1, 3, 0), (1, 3, 1), (0, 3, 1), (0, 3, 0), (0, 4, 0), (0, 5, 0), (0, 6, 0), (0,
7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9,
5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7),
(0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0,
6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8,
1), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9,
5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7),
(0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0,
6, 5), (0, 5, 5), (0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0),
(0, 5, 0), (0, 4, 0), (0, 4, 1), (0, 4, 2), (0, 4, 3), (0, 4, 4), (0, 4, 5), (0,
4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2,
9), (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 2, 4), (0, 2, 3), (0, 1, 3),
(0, 1, 4), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8), (0,
1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6)]

```

El coste total es de 281

La cantidad de nodos expandidos es de 300

El tiempo total de ejecucion es: 0.008786439895629883

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```

['Estacion C', '=', '=', '=', '#', '#', '=', '=', '=', 'E']
[ '=', 'Estanteria A', '#', '=', 'T', '=', '=', '#', '=', '=']
[ '=', '=', '#', '=', '=', '=', 'Estanteria B', '=', 'T', '=']
[ '=', 'E', '#', '#', '=', '#', '=', '=', '=', '=']
[ '=', '=', '=', '=', '=', '=', '#', 'Estacion U', '=']
[ '=', '#', '=', '=', 'E', '=', '*', '#', '=']
[ '=', '=', 'T', '=', '#', '=', '=', '=', '=', '=']
[ '=', '#', '=', 'Estacion R', '=', '=', '#', '=', '=', 'Estanteria X', '=']
[ '=', '=', '=', '#', '*', '#', '*', '=', '=', '=']
[ '=', '=', '=', 'T', '=', '=', '=', '=', '=', '=']

```

```

[ '=', '=', '=', '=', '=', 'Estanteria J', '=', '=', 'E']
[ '=', 'Estanteria D', '=', 'T', '.', '.', '.', '.', '.', '.']
[ '=', '*', '#', '=', '=', '.', '.', '#', '.', '.']
[ '=', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
[ '=', '=', '=', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '=', '=', '#', '=', 'E', '=', '*', '#', '.', '.']
[ '=', '=', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '=']
[ '=', '#', '=', '=', '=', '#', '=', 'Estanteria I', '=']
[ '=', '*', '#', '.', '#', '*', '=', '=', '=']
[ '=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=']
----- El mapa resultante del algoritmo DFS Tree es -----

```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: []

El coste total es de 0

La cantidad de nodos expandidos es de 0

El tiempo total de ejecucion es: 0

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']

```

```
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '.', '.', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- COMPARATIVA FINAL DE LOS ALGORITMOS -----

```

Algoritmo	Tiempo Total (s)	Nodos Expandidos Totales	Coste Total
A* Graph Manhattan	0.003319978713989258	43	41
A* Graph Euclidea	0.0040400028228759766	47	41
A* Tree Manhattan	0.00551605224609375	86	41
A* Tree Euclidea	0.004470348358154297	99	41
Greedy Graph Manhattan	0.0007808208465576172	29	41
Greedy Graph Euclidea	0.0012278556823730469	29	41
Greedy Tree Manhattan	0.0020513534545898438	30	41
Greedy Tree Euclidea	0.0020072460174560547	29	41
BFS Graph	0.011821508407592773	330	41
BFS Tree	0	0	0
DFS Graph	0.008786439895629883	300	281
DFS Tree	0	0	0

EL PROBLEMA NO TIENE SOLUCION: No se ha encontrado ruta desde Estacion U a Estanteria F

----- El mapa resultante del algoritmo A* Graph Manhattan es -----

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 43

El tiempo total de ejecucion es: 0.003319978713989258

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
['.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
['.', '#', '.', '=', '=', 'E', '=', '.', '#', '*']
['.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
```

```
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '*', '*', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo A* Graph Euclidea es -----

-----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 47

El tiempo total de ejecucion es: 0.0040400028228759766

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '*', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '*']
[ '.', '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '.', '=', 'E', '=', '.', '#', '*']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '*', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

-----
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '*', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '*', '=', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo A* Tree Manhattan es -----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 86

El tiempo total de ejecucion es: 0.00551605224609375

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.', '.']
[ '.', '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '*']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '*', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo A* Tree Euclidea es -----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 99

El tiempo total de ejecucion es: 0.004470348358154297

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '*', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.', '.']
[ '.', '.', '.', '.', '.', '*', '=', '#', 'Estacion U', '*']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '*']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '*', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

```

----- El mapa resultante del algoritmo Greedy Graph Manhattan es
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),
(0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1,
4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7,
3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5),
(1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 41
La cantidad de nodos expandidos es de 29
El tiempo total de ejecucion es: 0.0007808208465576172
----- CAMINO RECORRIDO -----
----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria
J'] ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '.', '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo Greedy Graph Euclidea es -
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),
(0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1,
4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7,
3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5),
(1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 41
La cantidad de nodos expandidos es de 29
El tiempo total de ejecucion es: 0.0012278556823730469
----- CAMINO RECORRIDO -----
----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria
J'] ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '.', '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']

```

```
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo Greedy Tree Manhattan es -
-----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),
(0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1,
4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7,
3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5),
(1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 30

El tiempo total de ejecucion es: 0.0020513534545898438

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
[ '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

-----
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo Greedy Tree Euclidea es --
-----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6),
(0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 5, 4), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1,
4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7,
3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5),
(1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 29

El tiempo total de ejecucion es: 0.0020072460174560547

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '=', '=', '.', '.']
[ '.', '.', '.', '.', '.', '=', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '=', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '=', '=', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo BFS Graph es -----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----
 El camino recorrido es: [(0, 4, 8), (0, 3, 8), (0, 3, 7), (0, 3, 6), (0, 4, 6), (0, 5, 6), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 3), (1, 4, 4), (1, 4, 5), (1, 5, 5), (0, 5, 5), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 41

La cantidad de nodos expandidos es de 330

El tiempo total de ejecucion es: 0.011821508407592773

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '*', '*', '*', '#', '#', '*', '*', '*', 'E']
[*', 'Estanteria A', '#', '*', 'T', '*', '*', '#', '*', '*']
[*', '*', '#', '*', '*', 'Estanteria B', '*', 'T', '*']
[*', 'E', '#', '#', '*', '#', '=', '=', '=', '*', '*']
[*', '*', '*', '*', '*', '=', '#', 'Estacion U', '*']
[*', '#', '*', '=', '=', 'E', '=', '*', '#', '*']
[*', '*', 'T', '=', '#', '=', '*', '*', '*', '*']
[*', '#', '*', 'Estacion R', '=', '=', '#', '*', 'Estanteria X', '*']
[*', '*', '*', '#', '*', '#', '*', '*', '*', '*']
[*', '*', '*', 'T', '*', '*', '*', '*', '*', '*']

[*', '*', '*', '*', '*', '*', 'Estanteria J', '*', '*', 'E']
```

```
[*, 'Estanteria D', '*', 'T', '*', '*', '=', '*', '*', '*']
[*', '*', '#', '*', '*', '*', '=', '#', '*', '*']
[*', 'E', '#', '#', '*', '#', '=', '*', '*', '*']
[*', '*', '*', 'Estanteria F', '=', '=', '=', '#', '*', '*']
[*', '*', '#', '*', '*', 'E', '*', '*', '#', '.']
[*', '*', 'T', '*', '#', '*', 'G', '*', 'Estanteria H', '*']
[*', '#', '*', '*', '*', '*', '#', '*', 'Estanteria I', '.']
[*', '*', '*', '#', '*', '#', '.', '*', '.', '.']
[., '*', '*', 'T', '*', '*', 'Estanteria K', '.', 'Estanteria L', '.']
```

----- El mapa resultante del algoritmo BFS Tree es -----

----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: []
El coste total es de 0
La cantidad de nodos expandidos es de 0
El tiempo total de ejecucion es: 0
----- CAMINO RECORRIDO -----
----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

----- El mapa resultante del algoritmo DFS Graph es -----

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 4, 8), (0, 4, 9), (0, 5, 9), (0, 6, 9), (0, 6, 8),
(0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 2), (0,
8, 2), (0, 8, 1), (0, 8, 0), (0, 7, 0), (0, 6, 0), (0, 6, 1), (0, 6, 2), (0, 6,
3), (0, 5, 3), (0, 5, 2), (0, 4, 2), (0, 4, 1), (0, 4, 0), (0, 3, 0), (0, 3, 1),
(0, 2, 1), (0, 2, 0), (0, 1, 0), (0, 1, 1), (0, 0, 1), (0, 0, 2), (0, 0, 3), (0,
1, 3), (0, 2, 3), (0, 2, 4), (0, 3, 4), (0, 4, 4), (0, 5, 4), (0, 5, 5), (0, 5,
6), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 3, 9), (0, 2, 9), (0, 2, 8),
(0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0,
0, 8), (0, 1, 8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0,
6), (1, 0, 5), (1, 0, 4), (1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0),
(1, 2, 0), (1, 3, 0), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1,
9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9,
7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8),
(1, 7, 9), (1, 6, 9), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1,
7, 4), (1, 7, 3), (1, 7, 2), (1, 6, 2), (1, 6, 1), (1, 5, 1), (1, 4, 1), (1, 4,
2), (1, 4, 3), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 4, 0), (1, 5, 0), (1, 6, 0),
(1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1,
9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8,
7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 6, 8), (1, 6, 7), (1, 6, 6),
(1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7, 3), (1, 7, 2), (1, 6, 2), (1, 6, 3), (1,
5, 3), (1, 5, 4), (1, 5, 5), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 3,
4), (1, 2, 4), (1, 2, 3), (1, 1, 3), (1, 1, 2), (1, 1, 1), (1, 1, 0), (1, 2, 0),
(1, 3, 0), (1, 3, 1), (0, 3, 1), (0, 3, 0), (0, 4, 0), (0, 5, 0), (0, 6, 0), (0,
7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9,
5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7),

```
(0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 7, 5), (0, 7, 4), (0, 7, 3), (0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8, 1), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 5, 5), (0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 5, 0), (0, 4, 0), (0, 4, 1), (0, 4, 2), (0, 4, 3), (0, 4, 4), (0, 4, 5), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2, 9), (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 1, 4), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6)]
```

El coste total es de 281

La cantidad de nodos expandidos es de 300

El tiempo total de ejecucion es: 0.008786439895629883

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '=', '=', '#', '#', '=', '=', 'E']
[ '=', 'Estanteria A', '#', '=', 'T', '=', '=', '#', '=', '=']
[ '=', '=', '#', '=', '=', '=', 'Estanteria B', '=', 'T', '=']
[ '=', 'E', '#', '#', '=', '#', '=', '=', '=', '=']
[ '=', '=', '=', '=', '=', '=', '#', 'Estacion U', '=']
[ '=', '#', '=', '=', 'E', '=', '*', '#', '=']
[ '=', '=', 'T', '=', '#', '=', '=', '=', '=', '=']
[ '=', '#', '=', 'Estacion R', '=', '=', '#', '=', 'Estanteria X', '=']
[ '=', '=', '#', '*', '#', '*', '=', '=', '=', '=']
[ '=', '=', '=', 'T', '=', '=', '=', '=', '=', '=', '=']

[ '=', '=', '=', '=', '=', 'Estanteria J', '=', '=', 'E']
[ '=', 'Estanteria D', '=', 'T', '.', '.', '.', '.', '.', '.']
[ '=', '*', '#', '=', '.', '.', '.', '#', '.', '.']
[ '=', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
[ '=', '=', '=', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '=', '=', '#', '=', 'E', '=', '*', '#', '.']
[ '=', '=', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '=']
[ '=', '#', '=', '=', '=', '#', '=', 'Estanteria I', '=']
[ '=', '*', '*', '#', '.', '#', '*', '=', '=', '=']
[ '=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=']
```

----- El mapa resultante del algoritmo DFS Tree es -----

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: []

El coste total es de 0

La cantidad de nodos expandidos es de 0

El tiempo total de ejecucion es: 0

----- CAMINO RECORRIDO -----

----- Salida: Estacion U Destinos: ['Estanteria F', 'Estacion R', 'Estanteria J'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

----- COMPARATIVA FINAL DE LOS ALGORITMOS -----

Algoritmo	Tiempo Total (s)	Nodos Expandidos Totales	Coste Total
A* Graph Manhattan	0.003319978713989258	43	41
A* Graph Euclidea	0.0040400028228759766	47	41
A* Tree Manhattan	0.0051605224609375	86	41
A* Tree Euclidea	0.004470348358154297	99	41
Greedy Graph Manhattan	0.0007808208465576172	29	41
Greedy Graph Euclidea	0.0012278556823730469	29	41
Greedy Tree Manhattan	0.0020513534545898438	30	41
Greedy Tree Euclidea	0.0020072460174560547	29	41
BFS Graph	0.011821508407592773	330	41
BFS Tree	0	0	0
DFS Graph	0.008786439895629883	300	281
DFS Tree	0	0	0

A Graph (Manhattan)* es el algoritmo más eficiente en cuanto a tiempo, seguido de Greedy Graph, que es aún más rápido pero no garantiza el camino óptimo.

Los algoritmos A Tree* son mucho más lentos debido a la expansión repetida de nodos.

BFS y DFS tienen tiempos elevados y una expansión de nodos considerablemente mayor, lo que los hace menos eficientes en este caso.

Caso 3: Recoge un paquete en la planta 0, lo lleva a 2 estaciones en la planta 1 , vuelve a la 0

```
In [135...]: analisis("Estacion R",[ "Estanteria J","Estanteria H","Estacion C"])
```

```

Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: None
    Acción: [(0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 15
    F(n): 15
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: .
    Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 5, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 1
    Coste Acumulado: 15
    Heuristica: 0
    F(n): 15
    Profundidad: 11
)
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 15
La cantidad de nodos expandidos es de 15
----- CAMINO RECORRIDO -----
----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: None
)

```

```

Acción: [(1, 0, 6)]
Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
Coste: 0
Coste Acumulado: 0
Heuristica: 8
F(n): 8
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 8
    Heuristica: 0
    F(n): 8
    Profundidad: 8
)
El camino recorrido es: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
El coste total es de 8
La cantidad de nodos expandidos es de 17
----- CAMINO RECORRIDO -----
----- Salida: Estanteria J (1, 0, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '*', '*', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '*', '*', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '*', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '*', '*', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '=', '#', '*', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
    Coste Acumulado: 0
)

```

```

Heuristica: 19
F(n): 19
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estacion C(
    Estado: Estacion C
    Coordenadas: (0, 0, 0)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
    Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],
    Coste: 1
    Coste Acumulado: 19
    Heuristica: 0
    F(n): 19
    Profundidad: 15
)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
El coste total es de 19
La cantidad de nodos expandidos es de 41
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estacion C (0, 0, 0) -----

['Estacion C', '=', '*', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '.', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '*', '*', '*', '.', '#', 'Estacion U', '.']
['.', '#', '*', '*', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '*', '#', '*', '.', '.', '.']
['.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '.', '#', '*', '*', 'E', '=', '=', '#', '.']
['.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
El coste total es de 42
La cantidad de nodos expandidos es de 73
El tiempo total de ejecucion es: 0.004893779754638672

```

```

----- CAMINO RECORRIDO -----
----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

['Estacion C', '=', '*', '**', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '**', '.', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '**', '*', '**', '.', '#', 'Estacion U', '.']
['.', '#', '*', '=', '=', 'E', '.', '.', '#', '.']
['.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '*', '**', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '*', '**', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '*', '*', '.']
['.', '.', '#', '.', '.', '.', '=', '#', '*', '.']
['.', 'E', '#', '#', '*', '#', '=', '*', '*', '.']
['.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '*', '.']
['.', '.', '#', '*', '**', 'E', '=', '=', '#', '.']
['.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: None
    Acción: [(0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 12.927446638338974
    F(n): 12.927446638338974
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: .
    Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 1
    Coste Acumulado: 15
    Heuristica: 0.0
    F(n): 15.0
    Profundidad: 11
)
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 15
La cantidad de nodos expandidos es de 16
----- CAMINO RECORRIDO -----
----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

```

```

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
['.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
['.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
['.', '#', '*', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
['.', '.', '#', '.', '.', 'E', '*', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: None
    Acción: [(1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 6.324555320336759
    F(n): 6.324555320336759
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 8
    Heuristica: 0.0
    F(n): 8.0
    Profundidad: 8
)
El camino recorrido es: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
El coste total es de 8
La cantidad de nodos expandidos es de 20
----- CAMINO RECORRIDO -----
----- Salida: Estanteria J (1, 0, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.']

```

```
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '*', 'Estanteria J', '*', '*', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '*', '=', '*', '*', '.']
[ '.', '.', '#', '.', '.', '*', '=', '#', '*', '.']
[ '.', 'E', '#', '#', '#', '=', '*', '*', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '=', '#', '*', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 15.233345472033855
    F(n): 15.233345472033855
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estacion C(
    Estado: Estacion C
    Coordenadas: (0, 0, 0)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5), (0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
    Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],
    Coste: 1
    Coste Acumulado: 19
    Heuristica: 0.0
    F(n): 19.0
    Profundidad: 15
)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5), (0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
El coste total es de 19
La cantidad de nodos expandidos es de 57
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estacion C (0, 0, 0) -----

['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '*', '.', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '*', '.', '.', '.']
['*', '=', '=', '=', '=', '*', '#', 'Estacion U', '.']
[ '.', '#', '*', '*', '*', 'E', '*', '.', '#', '.']
```

```
[ '.', '.', 'T', '*', '#', '*', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '*', '#', '*', '.', '.', '.']
[ '.', '*', '*', 'Estanteria F', '*', '*', '*', '#', '.', '.']
[ '.', '#', '*', '*', 'E', '=', '*', '#', '*']
[ '.', '.', 'T', '*', '#', '*', 'G', '=', 'Estanteria H', '*']
[ '.', '#', '.', '*', '*', '#', '*', 'Estanteria I', '*']
[ '.', '.', '.', '#', '.', '.', '*', '*', '*', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5), (0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]

El coste total es de 42

La cantidad de nodos expandidos es de 93

El tiempo total de ejecucion es: 0.004608869552612305

----- CAMINO RECORRIDO -----

----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```
['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', 'E']
[*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
[*', '=', '#', '*', '*', '.', 'Estanteria B', '.', 'T', '.']
[*', 'E', '#', '#', '*', '#', '*', '.', '.', '.']
[*', '=', '=', '=', '=', '*', '#', 'Estacion U', '.']
['.', '#', '*', '=', '=', 'E', '*', '.', '#', '.']
['.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
['.', '#', '*', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
['.', '.', '.', '.', '.', '*', 'Estanteria J', '*', '*', 'E']
['.', 'Estanteria D', '.', 'T', '.', '*', '=', '*', '*', '.']
['.', '.', '#', '.', '.', '*', '=', '#', '*', '.']
['.', 'E', '#', '#', '*', '#', '=', '*', '*', '.']
['.', '*', '*', 'Estanteria F', '*', '=', '=', '#', '*', '.']
['.', '.', '#', '*', '*', 'E', '=', '=', '#', '*']
['.', '.', 'T', '*', '#', '*', 'G', '=', 'Estanteria H', '*']
['.', '#', '.', '*', '*', '*', '#', '*', 'Estanteria I', '*']
['.', '.', '.', '#', '.', '#', '.', '*', '*', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estacion R

Estado: Estacion R

Coordenadas: (0, 7, 3)

Padre: None

Acción: [(0, 7, 3)]

Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 15

F(n): 15

```

Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: .
    Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 5, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 1
    Coste Acumulado: 15
    Heuristica: 0
    F(n): 15
    Profundidad: 11
)
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 15
La cantidad de nodos expandidos es de 31
----- CAMINO RECORRIDO -----
----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

[['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']]
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: None
    Acción: [(1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 8
    F(n): 8
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----

```

El nodo resultante es:

Nodo Estanteria H(
 Estado: Estanteria H
 Coordenadas: (1, 6, 8)
 Padre: .
 Acción: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
 Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
 Coste: 1
 Coste Acumulado: 8
 Heuristica: 0
 F(n): 8
 Profundidad: 8
)

El camino recorrido es: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]

El coste total es de 8

La cantidad de nodos expandidos es de 29

----- CAMINO RECORRIDO -----

----- Salida: Estanteria J (1, 0, 6) Destino: Estanteria H (1, 6, 8) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '*', '*', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '*', '*', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '*', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '*', '*', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '=', '#', '*', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estanteria H(

Estado: Estanteria H
 Coordenadas: (1, 6, 8)
 Padre: None
 Acción: [(1, 6, 8)]
 Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
 Coste: 0
 Coste Acumulado: 0
 Heuristica: 19
 F(n): 19
 Profundidad: 0
)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estacion C(
 Estado: Estacion C
 Coordenadas: (0, 0, 0)

Padre: .

Acción: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]

Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],

Coste: 1

Coste Acumulado: 19

Heuristica: 0

F(n): 19

Profundidad: 15

)

El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]

El coste total es de 19

La cantidad de nodos expandidos es de 459

----- CAMINO RECORRIDO -----

----- Salida: Estanteria H (1, 6, 8) Destino: Estacion C (0, 0, 0) -----

```
['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '*', '.', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '**', '*', '*', '.', '#', 'Estacion U', '.']
['.', '#', '**', '*', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '*', '#', '*', '.', '.', '.']
['.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '.', '#', '*', '*', 'E', '=', '=', '#', '.']
['.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----
```

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]

El coste total es de 42

La cantidad de nodos expandidos es de 519

El tiempo total de ejecucion es: 0.07443594932556152

----- CAMINO RECORRIDO -----

----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```
['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '*', '.', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '**', '*', '*', '.', '#', 'Estacion U', '.']
['.', '#', '**', '=', 'E', '.', '.', '#', '.']
```

```
[ '.', '.', 'T', '=', '#', '*', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '*', '*', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=' , '*', '*', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '*', '.']
[ '.', 'E', '#', '#', '*', '#', '=', '*', '*', '.']
[ '.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '*', '.']
[ '.', '.', '#', '*', '*', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: None
    Acción: [(0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 12.927446638338974
    F(n): 12.927446638338974
    Profundidad: 0
)
```

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

```
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: .
    Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 1
    Coste Acumulado: 15
    Heuristica: 0.0
    F(n): 15.0
    Profundidad: 11
)
```

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 15

La cantidad de nodos expandidos es de 35

----- CAMINO RECORRIDO -----

----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
[ '.', '#', '*', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: None
    Acción: [(1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 6.324555320336759
    F(n): 6.324555320336759
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 8
    Heuristica: 0.0
    F(n): 8.0
    Profundidad: 8
)
El camino recorrido es: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
El coste total es de 8
La cantidad de nodos expandidos es de 28
----- CAMINO RECORRIDO -----
----- Salida: Estanteria J (1, 0, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '*', 'Estanteria J', '*', '*', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '*', '=', '*', '*', '.']
[ '.', '.', '#', '.', '*', '=', '#', '*', '.']
```

```

[ '.', 'E', '#', '#', '.', '#', '=', '*', '*', '.', ]
[ '.', '.', '.', 'Estanteria F', '.', '.', '=', '#', '*', '.', ]
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 15.233345472033855
    F(n): 15.233345472033855
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estacion C(
    Estado: Estacion C
    Coordenadas: (0, 0, 0)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5), (0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
    Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],
    Coste: 1
    Coste Acumulado: 19
    Heuristica: 0.0
    F(n): 19.0
    Profundidad: 15
)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5), (0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
El coste total es de 19
La cantidad de nodos expandidos es de 1546
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estacion C (0, 0, 0) ----

['Estacion C', '=', '**', '**', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '**', '.', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '#', '*', '.', '.', '.']
['*', '=', '=', '=', '=', '*', '#', 'Estacion U', '.']
[ '.', '#', '**', '*', 'E', '*', '.', '#', '.']
[ '.', '.', 'T', '*', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '*', '#', '*', '.', '.', '.']
[ '.', '*', '**', 'Estanteria F', '*', '**', '*', '#', '.', '.']

```

```
[ '.', '.', '#', '*', '*', 'E', '=', '*', '#', '*' ]
[ '.', '.', 'T', '*', '#', '*', 'G', '=', 'Estanteria H', '*' ]
[ '.', '#', '.', '*', '*', '#', '*', 'Estanteria I', '*' ]
[ '.', '.', '.', '#', '.', '#', '.', '*', '*', '.' ]
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.' ]
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5),
(1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1,
0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6,
7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5),
(0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 3, 1), (0, 2, 1), (0,
1, 1), (0, 0, 1), (0, 0, 0)]
El coste total es de 42
La cantidad de nodos expandidos es de 1609
El tiempo total de ejecucion es: 0.6079733371734619
----- CAMINO RECORRIDO -----
----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion
C'] -----

['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '*', '.', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '*', '.', '.', '.']
['*', '=', '=', '=', '=', '*', '#', 'Estacion U', '.']
['.', '#', '*', '=', 'E', '*', '.', '#', '.']
['.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
['.', '#', '*', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '*', 'Estanteria J', '*', '*', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '*', '=', '*', '*', '.']
[ '.', '.', '#', '.', '.', '*', '=', '#', '*', '.']
[ '.', 'E', '#', '#', '*', '#', '=', '*', '*', '.']
[ '.', '*', '*', 'Estanteria F', '*', '=', '=', '#', '*', '.']
[ '.', '.', '#', '*', '*', 'E', '=', '=', '#', '*']
[ '.', '.', 'T', '*', '#', '*', 'G', '=', 'Estanteria H', '*' ]
[ '.', '#', '.', '*', '*', '#', '*', 'Estanteria I', '*' ]
[ '.', '.', '.', '#', '.', '#', '.', '*', '*', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: None
    Accion: [(0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 15
    F(n): 15
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: .
    Accion: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1,
```

```

4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
Coste: 1
Coste Acumulado: 15
Heuristica: 0
F(n): 0
Profundidad: 11
)
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5),
(1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 15
La cantidad de nodos expandidos es de 11
----- CAMINO RECORRIDO -----
----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

[['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E'],
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.'],
['.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.'],
['.', 'E', '#', '#', '.', '#', '.', '.', '.', '.'],
['.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.'],
['.', '#', '.', '=', '=', 'E', '.', '.', '#', '.'],
['.', '.', 'T', '=', '#', '.', '.', '.', '.', '.'],
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.'],
['.', '.', '.', '#', '.', '#', '.', '.', '.'],
['.', '.', 'T', '.', '.', '.', '.', '.', '.'],

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E'],
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.'],
['.', '.', '#', '.', '.', '.', '=', '#', '.', '.'],
['.', 'E', '#', '#', '.', '#', '=', '.', '.', '.'],
['.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.'],
['.', '.', '#', '.', '.', 'E', '.', '.', '#', '.'],
['.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.'],
['.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.'],
['.', '.', '.', '#', '.', '#', '.', '.', '.'],
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: None
    Acción: [(1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 8
    F(n): 8
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 0, 6), (1, 0, 7), (1, 0, 8), (1, 1, 8), (1, 2, 8), (1, 3, 8), (1,
4, 8), (1, 4, 9), (1, 5, 9), (1, 6, 9), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 10

```

```

Heuristica: 0
F(n): 0
Profundidad: 10
)
El camino recorrido es: [(1, 0, 6), (1, 0, 7), (1, 0, 8), (1, 1, 8), (1, 2, 8),
(1, 3, 8), (1, 4, 8), (1, 4, 9), (1, 5, 9), (1, 6, 9), (1, 6, 8)]
El coste total es de 10
La cantidad de nodos expandidos es de 10
----- CAMINO RECORRIDO -----
----- Salida: Estanteria J (1, 0, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '=', '=', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '=', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '=', '.']
[ '.', 'E', '#', '#', '#', '.', '.', '=', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '=', '=']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '=']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '=']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 19
    F(n): 19
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estacion C(
    Estado: Estacion C
    Coordenadas: (0, 0, 0)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1,
    4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1,
    1), (0, 0, 1), (0, 0, 0)]
    Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],
    Coste: 1
    Coste Acumulado: 19
    Heuristica: 0
    F(n): 0
    Profundidad: 15

```

```

)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6),
(1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0,
2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
El coste total es de 19
La cantidad de nodos expandidos es de 17
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estacion C (0, 0, 0) -----

['Estacion C', '=', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.']
['.', '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '*', '#', '*', '.', '.', '.']
['.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '.', '.']
['.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
['.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5),
(1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1,
0, 6), (1, 0, 7), (1, 0, 8), (1, 1, 8), (1, 2, 8), (1, 3, 8), (1, 4, 8), (1, 4,
9), (1, 5, 9), (1, 6, 9), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6),
(1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0,
3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
El coste total es de 44
La cantidad de nodos expandidos es de 38
El tiempo total de ejecucion es: 0.0012750625610351562
----- CAMINO RECORRIDO -----
----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion
C'] -----

['Estacion C', '=', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.']
['.', '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
['.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
['.', '.', 'T', '.', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '=', '=', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '=', '.']
['.', '.', '#', '.', '.', '.', '=', '#', '=', '.']
['.', 'E', '#', '#', '*', '#', '=', '.', '=', '.']
['.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '=', '.']
```

```
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '=']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '=']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: None
    Acción: [(0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 12.927446638338974
    F(n): 12.927446638338974
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: .
    Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 1
    Coste Acumulado: 15
    Heuristica: 0.0
    F(n): 0.0
    Profundidad: 11
)
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 15
La cantidad de nodos expandidos es de 11
----- CAMINO RECORRIDO -----
----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
```

```
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: None
    Acción: [(1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 6.324555320336759
    F(n): 6.324555320336759
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 8
    Heuristica: 0.0
    F(n): 0.0
    Profundidad: 8
)
El camino recorrido es: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
El coste total es de 8
La cantidad de nodos expandidos es de 8
----- CAMINO RECORRIDO -----
----- Salida: Estanteria J (1, 0, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
```

```

Padre: None
Acción: [(1, 6, 8)]
Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
Coste: 0
Coste Acumulado: 0
Heuristica: 15.233345472033855
F(n): 15.233345472033855
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estacion C(
    Estado: Estacion C
    Coordenadas: (0, 0, 0)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5), (0, 4, 5), (0, 4, 4), (0, 3, 4), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 0, 3), (0, 0, 2), (0, 0, 1), (0, 0, 0)]
    Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],
    Coste: 1
    Coste Acumulado: 19
    Heuristica: 0.0
    F(n): 0.0
    Profundidad: 15
)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5), (0, 4, 5), (0, 4, 4), (0, 3, 4), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 0, 3), (0, 0, 2), (0, 0, 1), (0, 0, 0)]
El coste total es de 19
La cantidad de nodos expandidos es de 15
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estacion C (0, 0, 0) -----

[['Estacion C', '=', '=', '#', '#', '.', '.', '.', 'E'],
 ['.', 'Estanteria A', '#', '=', 'T', '.', '.', '#', '.', '.'],
 ['.', '.', '#', '=', '.', 'Estanteria B', '.', 'T', '.'],
 ['.', 'E', '#', '#', '=', '#', '.', '.', '.', '.'],
 ['.', '.', '.', '.', '=', '.', '#', 'Estacion U', '.'],
 ['.', '#', '.', '.', 'E', '.', '.', '#', '.'],
 ['.', '.', 'T', '.', '#', '.', '.', '.', '.'],
 ['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.'],
 ['.', '.', '.', '#', '.', '#', '.', '.', '.'],
 ['.', '.', '.', 'T', '.', '.', '.', '.', '.'],

 ['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E'],
 ['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.'],
 ['.', '.', '#', '.', '.', '.', '#', '.', '.'],
 ['.', 'E', '#', '#', '.', '.', '.', '.'],
 ['.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.'],
 ['.', '.', '#', '.', '.', 'E', '=', '.', '#', '.'],
 ['.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.'],
 ['.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.'],
 ['.', '.', '#', '.', '#', '.', '.', '.'],
 ['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5),
]

```

(0, 4, 5), (0, 4, 4), (0, 3, 4), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 0, 3), (0, 0, 2), (0, 0, 1), (0, 0, 0)]

El coste total es de 42

La cantidad de nodos expandidos es de 34

El tiempo total de ejecucion es: 0.0008139610290527344

----- CAMINO RECORRIDO -----

----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```
['Estacion C', '=', '=', '=', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '=', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '=', '=', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '=', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estacion R

Estado: Estacion R

Coordenadas: (0, 7, 3)

Padre: None

Acción: [(0, 7, 3)]

Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 15

F(n): 15

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria J(

Estado: Estanteria J

Coordenadas: (1, 0, 6)

Padre: .

Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],

Coste: 1

Coste Acumulado: 15

Heuristica: 0

F(n): 0

Profundidad: 11

)

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 15
 La cantidad de nodos expandidos es de 11
 ----- CAMINO RECORRIDO -----
 ----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria J(
  Estado: Estanteria J
  Coordenadas: (1, 0, 6)
  Padre: None
  Acción: [(1, 0, 6)]
  Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
  Coste: 0
  Coste Acumulado: 0
  Heuristica: 8
  F(n): 8
  Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
  Estado: Estanteria H
  Coordenadas: (1, 6, 8)
  Padre: .
  Acción: [(1, 0, 6), (1, 0, 7), (1, 0, 8), (1, 1, 8), (1, 2, 8), (1, 3, 8), (1, 4, 8), (1, 4, 9), (1, 5, 9), (1, 6, 9), (1, 6, 8)]
  Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
  Coste: 1
  Coste Acumulado: 10
  Heuristica: 0
  F(n): 0
  Profundidad: 10
)
El camino recorrido es: [(1, 0, 6), (1, 0, 7), (1, 0, 8), (1, 1, 8), (1, 2, 8), (1, 3, 8), (1, 4, 8), (1, 4, 9), (1, 5, 9), (1, 6, 9), (1, 6, 8)]
El coste total es de 10
La cantidad de nodos expandidos es de 12
----- CAMINO RECORRIDO -----
----- Salida: Estanteria J (1, 0, 6) Destino: Estanteria H (1, 6, 8) -----
```

```

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '=', '=', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '=', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '=', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '=', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '=', '=']
[ '.', '.', '#', '.', 'E', '.', '.', '#', '=']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '=']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 19
    F(n): 19
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estacion C(
    Estado: Estacion C
    Coordenadas: (0, 0, 0)
    Padre: .
    Acción: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
    Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],
    Coste: 1
    Coste Acumulado: 19
    Heuristica: 0
    F(n): 0
    Profundidad: 15
)
El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
El coste total es de 19
La cantidad de nodos expandidos es de 17
----- CAMINO RECORRIDO -----
----- Salida: Estanteria H (1, 6, 8) Destino: Estacion C (0, 0, 0) -----
['Estacion C', '=', '.', '.', '#', '#', '.', '.', '.', 'E']

```

```
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '*', '#', '*', '.', '.', '.']
[ '.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- RESULTADO FINAL DE LA BUSQUEDA -----

```

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1, 0, 6), (1, 0, 7), (1, 0, 8), (1, 1, 8), (1, 2, 8), (1, 3, 8), (1, 4, 8), (1, 4, 9), (1, 5, 9), (1, 6, 9), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]

El coste total es de 44

La cantidad de nodos expandidos es de 40

El tiempo total de ejecucion es: 0.0008676052093505859

----- CAMINO RECORRIDO -----

----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```
['Estacion C', '=', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '=', '=', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '=', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '=', '.']
[ '.', 'E', '#', '#', '*', '#', '=', '.', '=', '.']
[ '.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '=', '=']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '=']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '=']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', 'T', '.', '.', '.', '.', '.', '.']

Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: None
    Accion: [(0, 7, 3)]
```

```

Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
Coste: 0
Coste Acumulado: 0
Heuristica: 12.927446638338974
F(n): 12.927446638338974
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: .
    Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 5, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 1
    Coste Acumulado: 15
    Heuristica: 0.0
    F(n): 0.0
    Profundidad: 11
)
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
El coste total es de 15
La cantidad de nodos expandidos es de 11
----- CAMINO RECORRIDO -----
----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: None
    Acción: [(1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 6.324555320336759

```

```

F(n): 6.324555320336759
Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)], Coste: 1
    Coste Acumulado: 8
    Heuristica: 0.0
    F(n): 0.0
    Profundidad: 8
)
El camino recorrido es: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
El coste total es de 8
La cantidad de nodos expandidos es de 8
----- CAMINO RECORRIDO -----
----- Salida: Estanteria J (1, 0, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: None
    Acción: [(1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)], Coste: 0
    Coste Acumulado: 0
    Heuristica: 15.233345472033855
    F(n): 15.233345472033855
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO

```

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estacion C(

Estado: Estacion C

Coordenadas: (0, 0, 0)

Padre: .

Acción: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5), (0, 4, 5), (0, 4, 4), (0, 3, 4), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 0, 3), (0, 0, 2), (0, 0, 1), (0, 0, 0)]

Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],

Coste: 1

Coste Acumulado: 19

Heuristica: 0.0

F(n): 0.0

Profundidad: 15

)

El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5), (0, 4, 5), (0, 4, 4), (0, 3, 4), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 0, 3), (0, 0, 2), (0, 0, 1), (0, 0, 0)]

El coste total es de 19

La cantidad de nodos expandidos es de 15

----- CAMINO RECORRIDO -----

----- Salida: Estanteria H (1, 6, 8) Destino: Estacion C (0, 0, 0) -----

```
['Estacion C', '=', '=', '=', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '=', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '=', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '=', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5), (0, 4, 5), (0, 4, 4), (0, 3, 4), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 0, 3), (0, 0, 2), (0, 0, 1), (0, 0, 0)]

El coste total es de 42

La cantidad de nodos expandidos es de 34

El tiempo total de ejecucion es: 0.0017621517181396484

----- CAMINO RECORRIDO -----

----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```
['Estacion C', '=', '=', '=', '#', '#', '.', '.', '.', 'E']
```

```
[ '.', 'Estanteria A', '#', '=', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '=', '=', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '=', '=', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=' , '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '=', '=', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estacion R(

Estado: Estacion R
 Coordenadas: (0, 7, 3)
 Padre: None
 Acción: [(0, 7, 3)]
 Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
 Coste: 0
 Coste Acumulado: 0
 Heuristica: 15
 F(n): 15
 Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estanteria J(

Estado: Estanteria J
 Coordenadas: (1, 0, 6)
 Padre: .
 Acción: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]
 Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
 Coste: 1
 Coste Acumulado: 15
 Heuristica: 0
 F(n): 0
 Profundidad: 11

)

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6)]

El coste total es de 15

La cantidad de nodos expandidos es de 134

----- CAMINO RECORRIDO -----

----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

```
['Estacion C', '*', '*', '*', '#', '#', '*', '*', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '*', '*', '#', '*', '.', '.']
['*', '*', '#', '*', '*', '*', 'Estanteria B', '*', 'T', '*']
['*', 'E', '#', '#', '*', '#', '*', '*', '*', '*']
['*', '*', '*', '*', '*', '*', '#', 'Estacion U', '*']
```

```
[ '*', '#', '*', '=', '=', 'E', '*', '*', '#', '*' ]
[ '*', '*', 'T', '=', '#', '*', '*', '*', '*', '*' ]
[ '*', '#', '*', 'Estacion R', '*', '*', '#', '*', 'Estanteria X', '*' ]
[ '*', '*', '#', '*', '#', '*', '*', '*', '*' ]
[ '*', '*', '*', 'T', '*', '*', '*', '*', '*', '*' ]

[ '*', '*', '*', '.', '.', '.', 'Estanteria J', '.', '.', 'E' ]
[ '*', 'Estanteria D', '*', 'T', '*', '.', '=', '.', '.', '.']
[ '*', '*', '#', '*', '*', '*', '=', '#', '.', '.']
[ '*', 'E', '#', '#', '*', '#', '=', '*', '*', '.']
[ '*', '*', '*', 'Estanteria F', '*', '=', '=', '#', '.', '.']
[ '*', '*', '#', '*', '*', 'E', '*', '*', '#', '.']
[ '*', '*', 'T', '*', '#', '*', 'G', '*', 'Estanteria H', '*' ]
[ '.', '#', '*', '*', '*', '#', '*', 'Estanteria I', '.']
[ '.', '.', '.', '#', '*', '#', '.', '*', '.', '.']
[ '.', '.', '.', 'T', '*', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: None
    Acción: [(1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 8
    F(n): 8
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 8
    Heuristica: 0
    F(n): 0
    Profundidad: 8
)
El camino recorrido es: [(1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8)]
El coste total es de 8
La cantidad de nodos expandidos es de 60
----- CAMINO RECORRIDO -----
----- Salida: Estanteria J (1, 0, 6) Destino: Estanteria H (1, 6, 8) -----

[ 'Estacion C', '.', '.', '.', '#', '#', '*', '*', '*', 'E' ]
[ '.', 'Estanteria A', '#', '.', 'T', '.', '*', '#', '*', '*' ]
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '*', 'T', '*' ]
[ '.', 'E', '#', '#', '.', '#', '.', '.', '*', '*' ]
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '*' ]
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '.', '.', '.', '.']
```

```
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.', '.']

[ '*', '*', '*', '*', '*', '*', 'Estanteria J', '*', '*', '*', 'E']
[ '*', 'Estanteria D', '*', 'T', '*', '*', '=', '*', '*', '*', '*']
[ '.', '*', '#', '*', '*', '=', '#', '*', '*', '*']
[ '.', 'E', '#', '#', '*', '#', '=', '*', '*', '*']
[ '.', '.', '.', 'Estanteria F', '*', '*', '=', '#', '*', '*']
[ '.', '.', '#', '.', '*', 'E', '=', '=', '#', '*']
[ '.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

Nodo Estanteria H(

Estado: Estanteria H

Coordenadas: (1, 6, 8)

Padre: None

Acción: [(1, 6, 8)]

Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 19

F(n): 19

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estacion C(

Estado: Estacion C

Coordenadas: (0, 0, 0)

Padre: .

Acción: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]

Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],

Coste: 1

Coste Acumulado: 19

Heuristica: 0

F(n): 0

Profundidad: 15

)

El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]

El coste total es de 19

La cantidad de nodos expandidos es de 170

----- CAMINO RECORRIDO -----

----- Salida: Estanteria H (1, 6, 8) Destino: Estacion C (0, 0, 0) -----

```
['Estacion C', '=', '*', '*', '#', '#', '*', '*', '*', 'E']
[ '*', 'Estanteria A', '#', '*', 'T', '*', '*', '#', '*', '*']
[ '*', '=', '#', '*', '*', 'Estanteria B', '*', 'T', '*']
[ '*', 'E', '#', '#', '*', '#', '*', '*', '*', '*']
[ '*', '*', '*', '*', '*', '*', '#', 'Estacion U', '*']
[ '*', '#', '*', '*', '*', 'E', '*', '*', '#', '*']
[ '*', '*', 'T', '*', '#', '*', '*', '*', '*', '*']
[ '*', '#', '*', 'Estacion R', '*', '*', '#', '*', 'Estanteria X', '*']
[ '*', '*', '#', '*', '#', '*', '*', '*', '*', '*']
[ '*', '*', '*', 'T', '*', '*', '*', '*', '*', '*']
```

```
[ '*', '*', '*', '*', '*', '*', 'Estanteria J', '*', '*', 'E']
[ '*', 'Estanteria D', '*', 'T', '*', '*', '*', '*', '*', '*']
[ '*', '*', '#', '*', '*', '*', '#', '*', '*', '*']
[ '*', 'E', '#', '#', '*', '#', '*', '*', '*', '*']
[ '*', '=', '=', 'Estanteria F', '=', '=', '=', '#', '*', '*']
[ '*', '#', '*', '*', 'E', '=', '=', '#', '*']
[ '*', '*', 'T', '*', '#', '*', 'G', '=', 'Estanteria H', '*']
[ '#', '*', '*', '*', '*', '#', '*', 'Estanteria I', '*']
[ '*', '*', '#', '*', '#', '*', '*', '*', '*', '*']
[ '*', '*', '*', 'T', '*', '*', 'Estanteria K', '*', 'Estanteria L', '*']

----- RESULTADO FINAL DE LA BUSQUEDA -----
```

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]

El coste total es de 42

La cantidad de nodos expandidos es de 364

El tiempo total de ejecucion es: 0.009092569351196289

----- CAMINO RECORRIDO -----

----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```
['Estacion C', '=', '*', '*', '#', '#', '*', '*', 'E']
[ '*', 'Estanteria A', '#', '*', 'T', '*', '*', '#', '*', '*']
[ '=', '#', '*', '*', '*', 'Estanteria B', '*', 'T', '*']
[ '*', 'E', '#', '#', '*', '#', '*', '*', '*', '*']
[ '*', '*', '*', '*', '*', '*', '#', 'Estacion U', '*']
[ '#', '*', '=', '=', 'E', '*', '*', '#', '*']
[ '*', 'T', '=', '#', '*', '*', '*', '*', '*']
[ '#', '*', 'Estacion R', '*', '*', '#', '*', 'Estanteria X', '*']
[ '*', '*', '#', '*', '#', '*', '*', '*', '*']
[ '*', '*', 'T', '*', '*', '*', '*', '*', '*']

[ '*', '*', '*', '*', '*', 'Estanteria J', '*', '*', 'E']
[ '*', 'Estanteria D', '*', 'T', '*', '*', '=', '*', '*', '*']
[ '*', '#', '*', '*', '*', '=', '#', '*', '*', '*']
[ '*', 'E', '#', '#', '*', '#', '=', '*', '*', '*']
[ '*', '=', '=', 'Estanteria F', '=', '=', '=', '#', '*', '*']
[ '*', '#', '*', '*', 'E', '=', '=', '#', '*']
[ '*', 'T', '*', '#', '*', 'G', '=', 'Estanteria H', '*']
[ '#', '*', '*', '*', '*', '#', '*', 'Estanteria I', '*']
[ '*', '*', '#', '*', '#', '*', '*', '*', '*', '*']
[ '*', '*', 'T', '*', '*', 'Estanteria K', '*', 'Estanteria L', '*']

----- Nodo Estacion R -----
```

Estado: Estacion R

Coordenadas: (0, 7, 3)

Padre: None

Acción: [(0, 7, 3)]

Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 15

F(n): 15

Profundidad: 0

)

EL RESULTADO NO PUEDE SER ENCONTRADO

EL PROBLEMA NO TIENE SOLUCION: No se ha encontrado ruta desde Estacion R a Estanteria J

```

Nodo Estacion R(
    Estado: Estacion R
    Coordenadas: (0, 7, 3)
    Padre: None
    Acción: [(0, 7, 3)]
    Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 15
    F(n): 15
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----

```

El nodo resultante es:

```

Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: .
    Acción: [(0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8, 1), (0, 8, 0), (0, 9, 0),
              (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7),
              (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8),
              (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 5, 5),
              (0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 5, 0),
              (0, 4, 0), (0, 4, 1), (0, 4, 2), (0, 4, 3), (0, 4, 4), (0, 4, 5), (0, 4, 6),
              (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2, 9),
              (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 2, 4), (0, 2, 3), (0, 1, 3),
              (0, 1, 4), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8),
              (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 1
    Coste Acumulado: 75
    Heuristica: 0
    F(n): 0
    Profundidad: 67
)
```

El camino recorrido es: [(0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8, 1), (0, 8, 0),
(0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0,
9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7,
8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 5, 5),
(0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 5, 0), (0,
4, 0), (0, 4, 1), (0, 4, 2), (0, 4, 3), (0, 4, 4), (0, 4, 5), (0, 4, 6), (0, 3,
6), (0, 3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2, 9), (0, 2, 8),
(0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 1, 4), (0,
1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8), (0, 1, 9), (0, 0,
9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6)]

El coste total es de 75

La cantidad de nodos expandidos es de 75

----- CAMINO RECORRIDO -----

----- Salida: Estacion R (0, 7, 3) Destino: Estanteria J (1, 0, 6) -----

```

['Estacion C', '.', '.', '.', '#', '#', '=', '=', '=', 'E']
['.', 'Estanteria A', '#', '=', 'T', '=', '=', '#', '=', '=']
['.', '.', '#', '=', '=', '=', 'Estanteria B', '=', 'T', '=']
['.', 'E', '#', '#', '.', '#', '=', '=', '=', '=']
['=', '=', '=', '=', '=', '=', '#', 'Estacion U', '=']
['=', '#', '=', '=', '=', 'E', '*', '*', '#', '*']
['=', '=', 'T', '.', '#', '=', '=', '=', '=', '=']
['*', '#', '=', 'Estacion R', '*', '*', '#', '=', 'Estanteria X', '=']
['=', '=', '#', '*', '#', '*', '=', '=', '=']

```

```
[ '=', '=', '=', 'T', '=', '=', '=', '=', '=', '=']
[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '=', '=', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

Nodo Estanteria J(
    Estado: Estanteria J
    Coordenadas: (1, 0, 6)
    Padre: None
    Acción: [(1, 0, 6)]
    Acciones disponibles: [((1, 0, 7), 1), ((1, 1, 6), 1), ((1, 0, 5), 1)],
    Coste: 0
    Coste Acumulado: 0
    Heuristica: 8
    F(n): 8
    Profundidad: 0
)
HAS LLEGADO AL OBJETIVO
----- RESULTADO DE LA BUSQUEDA -----
El nodo resultante es:
Nodo Estanteria H(
    Estado: Estanteria H
    Coordenadas: (1, 6, 8)
    Padre: .
    Acción: [(1, 0, 6), (1, 0, 5), (1, 0, 4), (1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 6, 8)]
    Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],
    Coste: 1
    Coste Acumulado: 33
    Heuristica: 0
    F(n): 0
    Profundidad: 32
)
El camino recorrido es: [(1, 0, 6), (1, 0, 5), (1, 0, 4), (1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 6, 8)]
El coste total es de 33
La cantidad de nodos expandidos es de 33
----- CAMINO RECORRIDO -----
----- Salida: Estanteria J (1, 0, 6) Destino: Estanteria H (1, 6, 8) -----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
```

```
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
```

```
[ '=', '=', '=', '=', '=', 'Estanteria J', '.', '.', 'E']
[ '=', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '=', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '=', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '=', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '=', '.', '#', '.', '.', 'E', '.', '.', '#', '.', '.']
[ '=', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '=']
[ '=', '#', '.', '.', '.', '.', '#', '=', 'Estanteria I', '=']
[ '=', '.', '.', '#', '.', '#', '*', '=', '=', '=', '=']
[ '=', '=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=']
```

Nodo Estanteria H(

Estado: Estanteria H

Coordenadas: (1, 6, 8)

Padre: None

Acción: [(1, 6, 8)]

Acciones disponibles: [((1, 6, 9), 1), ((1, 7, 8), 1), ((1, 6, 7), 1)],

Coste: 0

Coste Acumulado: 0

Heuristica: 19

F(n): 19

Profundidad: 0

)

HAS LLEGADO AL OBJETIVO

----- RESULTADO DE LA BUSQUEDA -----

El nodo resultante es:

Nodo Estacion C(

Estado: Estacion C

Coordenadas: (0, 0, 0)

Padre: .

Acción: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7, 3), (1, 7, 2), (1, 8, 2), (1, 8, 1), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 5, 9), (1, 4, 9), (1, 4, 8), (1, 3, 8), (1, 3, 7), (1, 3, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 6, 1), (1, 6, 2), (1, 6, 3), (1, 5, 3), (1, 5, 4), (1, 5, 5), (0, 5, 5), (0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 5, 6), (0, 4, 6), (0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 4, 0), (0, 3, 0), (0, 3, 1), (0, 2, 1), (0, 2, 0), (0, 1, 0), (0, 1, 1), (0, 0, 1), (0, 0, 0)]

Acciones disponibles: [((0, 0, 1), 1), ((0, 1, 0), 1)],

Coste: 1

Coste Acumulado: 101

Heuristica: 0

F(n): 0

Profundidad: 93

)

El camino recorrido es: [(1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7, 3), (1, 7, 2), (1, 8, 2), (1, 8, 1), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 5, 9), (1, 4, 9), (1, 4, 8), (1, 3, 8), (1, 3, 7), (1, 3, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 6, 1), (1, 6, 2), (1, 6, 3), (1, 5, 3), (1, 5, 4), (1, 5, 5), (0, 5, 5), (0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 5, 6), (0, 4, 6), (0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 4, 0), (0, 3, 0), (0, 3, 1), (0, 2, 1), (0, 2, 0), (0, 1, 0), (0, 1, 1), (0, 0, 1), (0, 0, 0)]

```
(0, 5, 5), (0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 5, 6), (0, 4, 6), (0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 4, 0), (0, 3, 0), (0, 3, 1), (0, 2, 1), (0, 2, 0), (0, 1, 0), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
```

El coste total es de 101

La cantidad de nodos expandidos es de 109

----- CAMINO RECORRIDO -----

----- Salida: Estanteria H (1, 6, 8) Destino: Estacion C (0, 0, 0) -----

```
['Estacion C', '=', '.', '.', '#', '#', '.', '.', '.', 'E']
['=', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['=', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['=', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
['=', '=', '=', '=', '=', '=', '#', 'Estacion U', '.']
['*', '#', '=', '=', '=', 'E', '=', '*', '#', '.']
['=', '=', 'T', '*', '#', '*', '=', '=', '=', '=']
['=', '#', '*', 'Estacion R', '*', '*', '#', '=', 'Estanteria X', '=']
['=', '*', '*', '#', '*', '#', '*', '=', '=', '=']
['=', '=', '=', 'T', '=', '=', '=', '=', '=', '=']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
['=', '=', '=', 'Estanteria F', '=', '=', '=', '#', '=', '=']
['=', '.', '#', '=', '=', 'E', '*', '*', '#', '=']
['=', '=', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '=']
['*', '#', '=', '=', '=', '#', '=', 'Estanteria I', '=']
['=', '=', '#', '.', '#', '*', '=', '=', '=']
['=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=']
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8, 1), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 5, 5), (0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 5, 0), (0, 4, 0), (0, 4, 1), (0, 4, 2), (0, 4, 3), (0, 4, 4), (0, 4, 5), (0, 4, 6), (0, 3, 6), (0, 3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2, 9), (0, 2, 8), (0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 1, 4), (0, 1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8), (0, 1, 9), (0, 0, 9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6), (1, 0, 5), (1, 0, 4), (1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7, 3), (1, 7, 2), (1, 8, 2), (1, 8, 1), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 5, 9), (1, 4, 9), (1, 4, 8), (1, 3, 8), (1, 3, 7), (1, 3, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1, 6, 1), (1, 6, 2), (1, 6, 3), (1, 5, 3), (1, 5, 4), (1, 5, 5), (0, 5, 5), (0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 7, 0), (0, 8, 0), (0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0, 9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 7, 7), (0, 7, 8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 5, 6), (0, 4, 6), (0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 4, 0), (0, 3, 0), (0,

3, 1), (0, 2, 1), (0, 2, 0), (0, 1, 0), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
 El coste total es de 209
 La cantidad de nodos expandidos es de 217
 El tiempo total de ejecucion es: 0.005264997482299805
 ----- CAMINO RECORRIDO -----
 ----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```
['Estacion C', '=', '.', '.', '#', '#', '=', '=', '=', 'E']
['=', 'Estanteria A', '#', '=', 'T', '=', '=', '#', '=', '=']
['=', '=', '#', '=', '=', '=', 'Estanteria B', '=', 'T', '=']
['=', 'E', '#', '#', '.', '#', '=', '=', '=', '=']
['=', '=', '=', '=', '=', '=', '#', 'Estacion U', '=']
['=', '#', '=', '=', '=', 'E', '=', '*', '#', '*']
['=', '=', 'T', '*', '#', '=', '=', '=', '=', '=']
['=', '#', '=', 'Estacion R', '*', '*', '#', '=', 'Estanteria X', '=']
['=', '=', '#', '*', '#', '*', '=', '=', '=', '=']
['=', '=', '=', 'T', '=', '=', '=', '=', '=', '=']

['=', '=', '=', '=', '=', 'Estanteria J', '=', '=', 'E']
['=', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
['=', '.', '#', '.', '.', '.', '.', '#', '.', '.']
['=', 'E', '#', '#', '.', '#', '=', '=', '=', '.']
['=', '=', '=', 'Estanteria F', '=', '=', '=', '#', '=', '=']
['=', '.', '#', '=', 'E', '*', '*', '#', '=']
['=', '=', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '=']
['=', '#', '=', '=', '=', '#', '=', 'Estanteria I', '=']
['=', '=', '#', '.', '#', '*', '=', '=', '=']
['=', '=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=']
```

Nodo Estacion R(

- Estado: Estacion R
- Coordenadas: (0, 7, 3)
- Padre: None
- Acción: [(0, 7, 3)]
- Acciones disponibles: [((0, 6, 3), 1), ((0, 7, 4), 1), ((0, 7, 2), 1)],
- Coste: 0
- Coste Acumulado: 0
- Heuristica: 15
- F(n): 15
- Profundidad: 0

)

EL RESULTADO NO PUEDE SER ENCONTRADO

EL PROBLEMA NO TIENE SOLUCION: No se ha encontrado ruta desde Estacion R a Estanteria J

----- El mapa resultante del algoritmo A* Graph Manhattan es -----

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]

El coste total es de 42

La cantidad de nodos expandidos es de 73

El tiempo total de ejecucion es: 0.004893779754638672

----- CAMINO RECORRIDO -----

----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```

['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '*', '.', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '*', '*', '*', '.', '#', 'Estacion U', '.']
['.', '#', '*', '=', '=', 'E', '.', '.', '#', '.']
['.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '*', '*', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '*', '*', '.']
['.', '.', '#', '.', '.', '.', '=', '#', '*', '.']
['.', 'E', '#', '#', '*', '#', '=', '*', '*', '.']
['.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '*', '.']
['.', '.', '#', '*', '*', 'E', '=', '=', '#', '.']
['.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

----- El mapa resultante del algoritmo A* Graph Euclidea es ----
-----
```

```

----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5),
(1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1,
0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6,
7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5),
(0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 3, 1), (0, 2, 1), (0,
1, 1), (0, 0, 1), (0, 0, 0)]
El coste total es de 42
La cantidad de nodos expandidos es de 93
El tiempo total de ejecucion es: 0.004608869552612305
----- CAMINO RECORRIDO -----
----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion
C'] -----
```

```

['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '*', '.', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '*', '.', '.', '.']
['*', '=', '=', '=', '*', '#', 'Estacion U', '.']
['.', '#', '=', '=', 'E', '*', '.', '#', '.']
['.', '.', 'T', '=', '#', '*', '.', '.', '.']
['.', '#', '*', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '*', 'Estanteria J', '*', '*', 'E']
['.', 'Estanteria D', '.', 'T', '.', '*', '=', '*', '*', '.']
['.', '.', '#', '.', '.', '*', '=', '#', '*', '.']
['.', 'E', '#', '#', '*', '#', '=', '*', '*', '.']
['.', '*', '*', 'Estanteria F', '*', '=', '=', '#', '*', '.']
['.', '.', '#', '*', '*', 'E', '=', '=', '#', '*']
['.', '.', 'T', '*', '#', '*', 'G', '=', 'Estanteria H', '*']
['.', '#', '.', '*', '*', '#', '*', '*', 'Estanteria I', '*']
['.', '.', '.', '#', '.', '#', '.', '*', '*', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

----- El mapa resultante del algoritmo A* Tree Manhattan es -----
-----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----
 El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5),
 (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1,
 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6,
 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5),
 (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0,
 1, 1), (0, 0, 1), (0, 0, 0)]

El coste total es de 42

La cantidad de nodos expandidos es de 519

El tiempo total de ejecucion es: 0.07443594932556152

----- CAMINO RECORRIDO -----

----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```
['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '*', '.', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '.', '.', '.', '.']
['*', '*', '*', '*', '*', '.', '#', 'Estacion U', '.']
['.', '#', '*', '=', '=', 'E', '.', '.', '#', '.']
['.', '.', 'T', '=', '#', '*', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '*', '*', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '*', '*', '.']
['.', '.', '#', '.', '.', '.', '=', '#', '*', '.']
['.', 'E', '#', '#', '*', '#', '=', '*', '*', '.']
['.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '*', '.']
['.', '.', '#', '*', '*', 'E', '=', '=', '#', '.']
['.', '.', 'T', '.', '#', '*', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

----- El mapa resultante del algoritmo A* Tree Euclidea es -----

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5),
 (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1,
 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6,
 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5),
 (0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 3, 1), (0, 2, 1), (0,
 1, 1), (0, 0, 1), (0, 0, 0)]

El coste total es de 42

La cantidad de nodos expandidos es de 1609

El tiempo total de ejecucion es: 0.6079733371734619

----- CAMINO RECORRIDO -----

----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```
['Estacion C', '=', '*', '*', '#', '#', '.', '.', '.', 'E']
['*', 'Estanteria A', '#', '*', 'T', '.', '.', '#', '.', '.']
['*', '=', '#', '*', '*', '.', 'Estanteria B', '.', 'T', '.']
['*', 'E', '#', '#', '*', '#', '*', '.', '.', '.']
['*', '=', '=', '=', '*', '#', 'Estacion U', '.']
['.', '#', '*', '=', '=', 'E', '*', '.', '#', '.']
['.', '.', 'T', '=', '#', '*', '.', '.', '.']
['.', '#', '*', 'Estacion R', '*', '*', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
```

```
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '*', 'Estanteria J', '*', '*', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '*', '=', '*', '*', '.']
[ '.', '.', '#', '.', '.', '*', '=', '#', '*', '.']
[ '.', 'E', '#', '#', '*', '#', '=', '*', '*', '.']
[ '.', '*', '*', 'Estanteria F', '*', '=', '=', '#', '*', '.']
[ '.', '.', '#', '*', 'E', '=', '=', '#', '*']
[ '.', '.', 'T', '*', '#', '*', 'G', '=', 'Estanteria H', '*']
[ '.', '#', '.', '*', '*', '#', '*', 'Estanteria I', '*']
[ '.', '.', '.', '#', '.', '#', '.', '*', '*', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo Greedy Graph Manhattan es
-----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1, 0, 6), (1, 0, 7), (1, 0, 8), (1, 1, 8), (1, 2, 8), (1, 3, 8), (1, 4, 8), (1, 4, 9), (1, 5, 9), (1, 6, 9), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6), (1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0, 3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]

El coste total es de 44

La cantidad de nodos expandidos es de 38

El tiempo total de ejecucion es: 0.0012750625610351562

----- CAMINO RECORRIDO -----

----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```
['Estacion C', '=', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

-----
```

```
['.', '.', '.', '.', '.', '.', 'Estanteria J', '=', '=', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '=', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '=', '.']
[ '.', 'E', '#', '#', '*', '#', '=', '.', '=', '.']
[ '.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '=', '=']
[ '.', '.', '#', '.', '.', 'E', '=', '=', '#', '=']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '=']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

----- El mapa resultante del algoritmo Greedy Graph Euclidea es -
-----
```

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5), (1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1, 0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6, 7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5), (0, 4, 5), (0, 4, 4), (0, 3, 4), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 0, 3), (0, 0, 2), (0, 0, 1), (0, 0, 0)]

El coste total es de 42

La cantidad de nodos expandidos es de 34

```

El tiempo total de ejecucion es: 0.0008139610290527344
----- CAMINO RECORRIDO -----
----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] ----

['Estacion C', '=', '=', '=', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '=', 'T', '.', '.', '#', '.', '.']
['.', '.', '#', '=', '=', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
['.', '.', '.', '.', '=', '=', '.', '#', 'Estacion U', '.']
['.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
['.', '.', 'T', '=', '#', '.', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=' , '.', '.', '.']
['.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
['.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
['.', '.', '.', 'Estanteria F', '.', '.', '=', '=', '#', '.', '.']
['.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
['.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
['.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']

----- El mapa resultante del algoritmo Greedy Tree Manhattan es -
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5),
(1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1,
0, 6), (1, 0, 7), (1, 0, 8), (1, 1, 8), (1, 2, 8), (1, 3, 8), (1, 4, 8), (1, 4,
9), (1, 5, 9), (1, 6, 9), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 5, 7), (1, 5, 6),
(1, 4, 6), (1, 4, 5), (1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 3, 1), (0,
3, 1), (0, 2, 1), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
El coste total es de 44
La cantidad de nodos expandidos es de 40
El tiempo total de ejecucion es: 0.0008676052093505859
----- CAMINO RECORRIDO -----
----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] ----

['Estacion C', '=', '.', '.', '#', '#', '.', '.', '.', 'E']
['.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
['.', '=', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
['.', 'E', '#', '#', '.', '#', '.', '.', '.']
['.', '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
['.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
['.', '.', 'T', '=', '#', '.', '.', '.', '.']
['.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
['.', '.', '.', '#', '.', '#', '.', '.', '.']
['.', '.', '.', 'T', '.', '.', '.', '.', '.']

['.', '.', '.', '.', '.', '.', 'Estanteria J', '=', '=', 'E']
['.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '=', '.']
['.', '.', '#', '.', '.', '.', '=', '#', '=', '.']
['.', 'E', '#', '#', '*', '#', '=', '.', '=', '.']
['.', '=', '=', 'Estanteria F', '=', '=', '=', '#', '=', '=']
['.', '.', '#', '.', '.', 'E', '=', '=', '#', '=']
['.', '.', 'T', '=', '#', '.', '.', '.', '.']

file:///C:/Users/jochoa/Desktop/grupo2_p4.html
169/175

```

```
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
----- El mapa resultante del algoritmo Greedy Tree Euclidea es --
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5),
(1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1,
0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6,
7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5),
(0, 4, 5), (0, 4, 4), (0, 3, 4), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 0, 3), (0,
0, 2), (0, 0, 1), (0, 0, 0)]
El coste total es de 42
La cantidad de nodos expandidos es de 34
El tiempo total de ejecucion es: 0.0017621517181396484
-----
----- CAMINO RECORRIDO -----
----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion
C'] ----

['Estacion C', '=', '=', '=', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '=', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '=', '=', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '=', '#', '.', '.', '.', '.']
[ '.', '.', '.', '=', '=', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '=', '=', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '=', '#', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '=', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '=', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '=', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '=', '=', '#', '.', '.']
[ '.', '.', '.', '#', '.', '.', 'E', '=', '=', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '=', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
----- El mapa resultante del algoritmo BFS Graph es --
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 7, 3), (0, 6, 3), (0, 5, 3), (0, 5, 4), (0, 5, 5),
(1, 5, 5), (1, 4, 5), (1, 4, 6), (1, 3, 6), (1, 2, 6), (1, 1, 6), (1, 0, 6), (1,
0, 6), (1, 1, 6), (1, 2, 6), (1, 3, 6), (1, 4, 6), (1, 5, 6), (1, 5, 7), (1, 6,
7), (1, 6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 5, 6), (1, 5, 5), (0, 5, 5),
(0, 4, 5), (0, 4, 4), (0, 3, 4), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 0, 3), (0,
0, 2), (0, 0, 1), (0, 0, 0)]
El coste total es de 42
La cantidad de nodos expandidos es de 364
El tiempo total de ejecucion es: 0.009092569351196289
-----
----- CAMINO RECORRIDO -----
----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion
C'] ----

['Estacion C', '=', '**', '**', '#', '#', '**', '**', '**', 'E']
[ '**', 'Estanteria A', '#', '**', 'T', '**', '**', '#', '**', '**']
[ '**', '=', '#', '**', '**', '**', 'Estanteria B', '**', 'T', '**']
[ '**', 'E', '#', '**', '#', '**', '**', '**', '**']
```

```
['*', '*', '*', '*', '*', '*', '#', 'Estacion U', '*']
['*', '#', '*', '=', '=', 'E', '*', '*', '#', '*']
['*', '*', 'T', '=', '#', '*', '*', '*', '*', '*']
['*', '#', '*', 'Estacion R', '*', '*', '#', '*', 'Estanteria X', '*']
['*', '*', '*', '#', '*', '#', '*', '*', '*', '*']
['*', '*', '*', 'T', '*', '*', '*', '*', '*', '*', '*']

['*', '*', '*', '*', '*', '*', 'Estanteria J', '*', '*', 'E']
['*', 'Estanteria D', '*', 'T', '*', '*', '=', '*', '*', '*']
['*', '*', '#', '*', '*', '=', '#', '*', '*', '*']
['*', 'E', '#', '#', '*', '#', '=', '*', '*', '*']
['*', '=', '=', 'Estanteria F', '=', '=', '=', '#', '*', '*']
['*', '#', '*', '*', 'E', '=', '=', '#', '*']
['*', '*', 'T', '*', '#', '*', 'G', '=', 'Estanteria H', '*']
['*', '#', '*', '*', '*', '#', '*', '#', '*', 'Estanteria I', '*']
['*', '*', '#', '*', '#', '*', '#', '*', '*', '*', '*']
['*', '*', '*', 'T', '*', '*', 'Estanteria K', '*', 'Estanteria L', '*']
----- El mapa resultante del algoritmo BFS Tree es -----
-----
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: []
El coste total es de 0
La cantidad de nodos expandidos es de 0
El tiempo total de ejecucion es: 0
----- CAMINO RECORRIDO -----
----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] ----

['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', 'E']
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']
----- El mapa resultante del algoritmo DFS Graph es -----
-----
```

```
----- RESULTADO FINAL DE LA BUSQUEDA -----
El camino recorrido es: [(0, 7, 3), (0, 7, 2), (0, 8, 2), (0, 8, 1), (0, 8, 0),
(0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0,
9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7,
8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 6, 5), (0, 5, 5),
(0, 5, 4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 5, 0), (0,
4, 0), (0, 4, 1), (0, 4, 2), (0, 4, 3), (0, 4, 4), (0, 4, 5), (0, 4, 6), (0, 3,
6), (0, 3, 7), (0, 3, 8), (0, 4, 8), (0, 4, 9), (0, 3, 9), (0, 2, 9), (0, 2, 8),
(0, 2, 7), (0, 2, 6), (0, 2, 5), (0, 2, 4), (0, 2, 3), (0, 1, 3), (0, 1, 4), (0,
```

```

1, 5), (0, 1, 6), (0, 0, 6), (0, 0, 7), (0, 0, 8), (0, 1, 8), (0, 1, 9), (0, 0,
9), (1, 0, 9), (1, 0, 8), (1, 0, 7), (1, 0, 6), (1, 0, 6), (1, 0, 5), (1, 0, 4),
(1, 0, 3), (1, 0, 2), (1, 0, 1), (1, 0, 0), (1, 1, 0), (1, 2, 0), (1, 3, 0), (1,
4, 0), (1, 5, 0), (1, 6, 0), (1, 7, 0), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9,
2), (1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9),
(1, 8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1,
6, 8), (1, 6, 8), (1, 6, 7), (1, 6, 6), (1, 6, 5), (1, 7, 5), (1, 7, 4), (1, 7,
3), (1, 7, 2), (1, 8, 2), (1, 8, 1), (1, 8, 0), (1, 9, 0), (1, 9, 1), (1, 9, 2),
(1, 9, 3), (1, 9, 4), (1, 9, 5), (1, 9, 6), (1, 9, 7), (1, 9, 8), (1, 9, 9), (1,
8, 9), (1, 8, 8), (1, 8, 7), (1, 7, 7), (1, 7, 8), (1, 7, 9), (1, 6, 9), (1, 5,
9), (1, 4, 9), (1, 4, 8), (1, 3, 8), (1, 3, 7), (1, 3, 6), (1, 4, 6), (1, 4, 5),
(1, 4, 4), (1, 4, 3), (1, 4, 2), (1, 4, 1), (1, 4, 0), (1, 5, 0), (1, 6, 0), (1,
6, 1), (1, 6, 2), (1, 6, 3), (1, 5, 3), (1, 5, 4), (1, 5, 5), (0, 5, 5), (0, 5,
4), (0, 5, 3), (0, 5, 2), (0, 6, 2), (0, 6, 1), (0, 6, 0), (0, 7, 0), (0, 8, 0),
(0, 9, 0), (0, 9, 1), (0, 9, 2), (0, 9, 3), (0, 9, 4), (0, 9, 5), (0, 9, 6), (0,
9, 7), (0, 9, 8), (0, 9, 9), (0, 8, 9), (0, 8, 8), (0, 8, 7), (0, 7, 7), (0, 7,
8), (0, 7, 9), (0, 6, 9), (0, 6, 8), (0, 6, 7), (0, 6, 6), (0, 5, 6), (0, 4, 6),
(0, 4, 5), (0, 4, 4), (0, 4, 3), (0, 4, 2), (0, 4, 1), (0, 4, 0), (0, 3, 0), (0,
3, 1), (0, 2, 1), (0, 2, 0), (0, 1, 0), (0, 1, 1), (0, 0, 1), (0, 0, 0)]
```

El coste total es de 209

La cantidad de nodos expandidos es de 217

El tiempo total de ejecucion es: 0.005264997482299805

CAMINO RECORRIDO

----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```
['Estacion C', '=', '.', '.', '#', '#', '=', '=', '=', 'E']  
[ '=', 'Estanteria A', '#', '=', 'T', '=', '=', '#', '=', '=']  
[ '=', '=', '#', '=', '=', '=', 'Estanteria B', '=', 'T', '=' ]  
[ '=', 'E', '#', '#', '.', '#', '=', '=', '=', '=' ]  
[ '=', '=', '=', '=', '=', '=', '=', '#', 'Estacion U', '=' ]  
[ '=', '#', '=', '=', '=', 'E', '=', '*', '#', '*' ]  
[ '=', '=', 'T', '*', '#', '=', '=', '=', '=', '=' ]  
[ '=', '#', '=', 'Estacion R', '*', '*', '#', '=', 'Estanteria X', '=' ]  
[ '=', '=', '#', '*', '#', '*', '=', '=', '=', '=' ]  
[ '=', '=', '=', 'T', '=', '=', '=', '=', '=', '=' ]
```

```
[ '=', '=', '=', '=', '=', 'Estanteria J', '=', '=', 'E']  
[ '=', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']  
[ '=', '.', '#', '.', '.', '.', '.', '#', '.', '.']  
[ '=', 'E', '#', '#', '.', '#', '=', '=', '=', '.', '.']  
[ '=', '=', '=', 'Estanteria F', '=', '=', '=', '=', '#', '=', '=']  
[ '=', '.', '#', '=', '=', 'E', '*', '*', '#', '=' ]  
[ '=', '=', 'T', '=', '#', '=', 'G', '=', 'Estanteria H', '=' ]  
[ '=', '#', '=', '=', '=', '=', '#', '=', 'Estanteria I', '=' ]  
[ '=', '=', '=', '#', '.', '#', '*', '=', '=', '=' ]  
[ '=', '=', '=', 'T', '=', '=', 'Estanteria K', '=', 'Estanteria L', '=', 'Estanteria M']
```

----- El mapa resultante del algoritmo DFS Tree es -----

----- RESULTADO FINAL DE LA BUSQUEDA -----

El camino recorrido es: []

El coste total es de 0

La cantidad de nodos expandidos es de 0

El tiempo total de ejecucion es: 0

CAMINO RECORRIDO

----- Salida: Estacion R Destinos: ['Estanteria J', 'Estanteria H', 'Estacion C'] -----

```
['Estacion C', '.', '.', '.', '#', '#', '.', '.', '.', '.', 'E']  
[ '.', 'Estanteria A', '#', '.', 'T', '.', '.', '#', '.', '.']
```

```
[ '.', '.', '#', '.', '.', '.', 'Estanteria B', '.', 'T', '.']
[ '.', 'E', '#', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', '.', '.', '.', '#', 'Estacion U', '.']
[ '.', '#', '.', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', '.', '.', '.', '.']
[ '.', '#', '.', 'Estacion R', '.', '.', '#', '.', 'Estanteria X', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', '.', '.', '.']

[ '.', '.', '.', '.', '.', '.', 'Estanteria J', '.', '.', 'E']
[ '.', 'Estanteria D', '.', 'T', '.', '.', '.', '.', '.', '.']
[ '.', '.', '#', '.', '.', '.', '.', '#', '.', '.']
[ '.', 'E', '#', '#', '.', '.', '.', '.', '.']
[ '.', '.', '.', 'Estanteria F', '.', '.', '.', '#', '.', '.']
[ '.', '.', '#', '.', '.', 'E', '.', '.', '#', '.']
[ '.', '.', 'T', '.', '#', '.', 'G', '.', 'Estanteria H', '.']
[ '.', '#', '.', '.', '.', '.', '#', '.', 'Estanteria I', '.']
[ '.', '.', '.', '#', '.', '#', '.', '.', '.']
[ '.', '.', '.', 'T', '.', '.', '.', 'Estanteria K', '.', 'Estanteria L', '.']
```

COMPARATIVA FINAL DE LOS ALGORITMOS			
Algoritmo	Tiempo Total (s)	Nodos Expandidos Totales	Coste Total
A* Graph Manhattan	0.004893779754638672	73	42
A* Graph Euclidea	0.004608869552612305	93	42
A* Tree Manhattan	0.07443594932556152	519	42
A* Tree Euclidea	0.6079733371734619	1609	42
Greedy Graph Manhattan	0.0012750625610351562	38	44
Greedy Graph Euclidea	0.0008139610290527344	34	42
Greedy Tree Manhattan	0.0008676052093505859	40	44
Greedy Tree Euclidea	0.0017621517181396484	34	42
BFS Graph	0.009092569351196289	364	42
BFS Tree	0	0	0
DFS Graph	0.005264997482299805	217	209
DFS Tree	0	0	0

A Graph Manhattan* es el más eficiente en cuanto a tiempo y nodos expandidos, mientras que A Tree* (tanto Manhattan como Euclídea) es significativamente más lento debido a la expansión repetitiva de nodos.

Greedy Graph es muy rápido, especialmente Greedy Graph Euclídea, que es el más rápido, con un tiempo de 0.0026 s y 34 nodos expandidos.

BFS y DFS siguen mostrando un rendimiento más bajo debido a su expansión más amplia. BFS Tree y DFS Tree no encuentran una solución en este caso.

No se hacen mas busquedas para que la extension del pdf no sea demasiada

Planificación

Reflexionad sobre si habéis seguido la planificación que definisteis en el Entregable CG.2. Explicad qué aspectos se cumplieron, cuáles no, y las razones que llevaron a cualquier desviación.

En general no hemos seguido la planificación inicial definida en el entregable CG.2, o al menos no de la forma prevista. Aunque la estructura de tareas estaba bien planteada, la

realidad del desarrollo ha sido bastante diferente.

Por un lado, empezamos más tarde de lo previsto, principalmente porque al inicio no entendíamos con claridad cómo iba a ser la entrega final ni el nivel de complejidad que tendría. Esto hizo que no distribuyéramos correctamente el tiempo y que algunas tareas que deberían haber sido tempranas se retrasaran bastante.

Además, la planificación asumía un reparto equilibrado entre diseño, pruebas, documentación y programación. Sin embargo, la realidad ha sido que aproximadamente el 80% del tiempo se ha dedicado exclusivamente a programar. El desarrollo del código resultó ser mucho más complejo de lo esperado: implementar correctamente todas las versiones de los algoritmos, manejar correctamente el mapa , depurar problemas...

Como consecuencia:

Algunas tareas del esquema inicial, como la documentación y la reflexión, tuvieron que concentrarse al final.

El testeo y la comparación de algoritmos también se realizaron más tarde de lo planificado.

En resumen, la planificación inicial no se cumplió prácticamente, no por falta de organización, sino porque la complejidad real del proyecto y la curva de entendimiento no estaban bien estimadas al principio. Una vez comprendido el alcance, reorientamos el esfuerzo a completar las partes críticas del programa, lo que explica la desviación respecto al plan teórico.

Principales dudas y dificultades encontrados en el desarrollo

Una de las primeras dudas fue comprender bien cómo recorrer el mapa utilizando coordenadas (fila, columna) y cómo traducir eso en movimientos URDL, ya que en un principio estaba pensado como con coordenadas x , y .Y en la lista al ser con filas y columnas funciona al revés.

Interpretar correctamente el punto donde se pedía visitar varios destinos de forma consecutiva fue una dificultad. Entender que había que encadenar búsquedas consecutivas, actualizando el origen en cada tramo, por suerte no hubo que cambiar en gran medida el diseño del programa.

Al principio, al generar las rutas en el mapa, se modificaba el mapa original porque estaba copiando listas por referencia. La dificultad estuvo en construir una copia profunda copiando celda por celda que costó un poco sacar

También ha sido muy complejo entender el funcionamiento de las heurísticas y aplicarlo de forma que permita usar el ascensor más apto

Ha costado mucho la parte de analizar todas las funciones juntas y llamar a todas y tener el resultado de todas ellas para hacer la tabla, al principio se hacia a lo bruto usando las

funciones multiples con todos los algoritmos y guardando en variables todos los resultados, al buscar una solucion menos repetitiva y mas limpia el uso del diccionario ha resultado ser una solucion muy buena y hace que sea facil manejar cuales se quieren incluir y tener los resultados de forma perfectamente accesible.

No he conseguido una manera para mostrar el mapa alineado si los nombres no son caracteres y ocupan todos lo mismo