



Ejercicio de Laboratorio 2. DFS

Ejercicio 1. 4-Puzzle

Cambios Realizados

Cambio 1: En el constructor existe un error de inicialización, teníamos que no tendría 'hijos' por defecto, y después se llama al método 'set_hijos'.

```
class Nodo:
    def __init__(self, datos, hijos=None):
        self.datos = datos
        self.hijos = None
        self.padre = None
        self.coste = None
        self.set_hijos(hijos)

class Nodo:
    # Constructor
    # @param
    # datos: datos del nodo
    # padre: nodo padre esta vacío por defecto
    def __init__(self, datos, padre=None):
        self.datos = datos
        self.padre = padre
        self.hijos = []
        self.coste = 0
```

Cambio 2: Se quito el 'else' cuando se revisa si ya se encontró la solución.

```
if nodo.get_datos() == solucion:
    # solución encontrada
    solucionado = True
    return nodo
else:
    # expandir nodos hijo

if nodo_actual.get_datos() == solucion:
    solucionado = True
    return nodo_actual

# Añadimos el nodo actual a la lista de nodos visitados
nodos_visitados.append(nodo_actual)
```

Cambio 3: Para expandir a los nodos 'hijos', se opto por crear una función que generara los estados 'hijos' y después se revisara si estos no estuvieran en 'nodos_visitados' y en 'nodos_frontera'.

```
# Generamos los hijos del nodo actual y los añadimos a la frontera
hijos = generar_hijos(nodo_actual)
for hijo in hijos:
    if (not hijo.en_lista(nodos_visitados)) and (not hijo.en_lista(nodos_frontera)):
        nodos_frontera.append(hijo)
```

La función 'generar_hijos' devuelve una lista con los nodos hijos.



INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
NOMBRE: Cerda García Gustavo



Programa en funcionamiento

```
Estado Inicial: [4, 2, 3, 1]
Resultado:
[[1, 2, 3, 4]]
Camino a la solución:
→ [4, 2, 3, 1]
→ [4, 2, 1, 3]
→ [4, 1, 2, 3]
→ [4, 1, 3, 2]
→ [4, 3, 1, 2]
→ [3, 4, 1, 2]
→ [3, 4, 2, 1]
→ [3, 2, 4, 1]
→ [3, 2, 1, 4]
→ [3, 1, 2, 4]
→ [1, 3, 2, 4]
→ [1, 2, 3, 4]
```

Ejercicio 2. Laberinto

Programa en funcionamiento

- Laberinto Normal con solución

```
Maze Solved!
11111
**101
1*101
1***E
11111
Tiempo de ejecucion: 0.0
```



INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
NOMBRE: Cerda García Gustavo



- Laberinto más grande con solución

Modificaciones:

1. El laberinto se modificó para que fuera de 13x13

```
Maze Solved!
1111111111111
****101010001
111*101010101
1**101010101
1*11101010101
1*10001010101
1*10100010101
1*10111110101
1*1000000101
1*11111110101
1*****10101
1111111*10101
1E*****10101
1111111111111
Tiempo de ejecucion: 0.0
```

- Laberinto complejo

Modificaciones:

1. El laberinto es grande (10x10)
2. Se añadieron más obstáculos y caminos estrechos.
3. Se movió la posición final a una ubicación diferente.

```
Maze Solved!
1111111111
**10101011
1*10101011
1*****1011
11101*1011
10101*1011
10101*1011
10000*1011
11111*1011
11111E1001
Tiempo de ejecucion: 0.0

[Done] exited with code=0 in 0.072 seconds
```



INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
NOMBRE: Cerda García Gustavo



- Laberinto sin solución

```
No solution found.  
Tiempo de ejecución: 0.0  
  
[Done] exited with code=0 in 0.065 seconds
```