

Sistemas Inteligentes

Proyecto 1

Búsquedas NO informadas

Antecedentes

El 8-puzzle es un juego que consta de 9X9 casilleros. Los casilleros contienen 8 fichas, cada una marcada con un número (ver figura 1), deslizables en forma horizontal o vertical y uno de los casilleros queda vacío. El juego permite deslizar alguna de las fichas en forma vertical y horizontal sobre el espacio en blanco siempre y cuando la ficha esté en un lugar 4-adyacente al espacio (4-adyacente significa que esté arriba, abajo, a la izquierda o a la derecha del espacio).

1	2	3
4	5	6
7	8	

Figura 1. Un estado del 8-puzzle.

El juego consiste en que a partir del estado inicial, mover las fichas sobre el blanco para tratar de ordenar los números hasta que se llegue al estado meta.

Definición

Hacer una función en Python (el archivo que la contenga deberá llamarse **Proyecto1.py**) llamada:

busquedaNoInformada(edoInicial, edoFinal, algoritmo)

que reciba como parámetros:

- Un estado inicial del 8-puzzle. El estado inicial es una lista de listas, donde cada lista interna contendrá 3 dígitos del 0 al 8 representa un renglón del 8-puzzle. El espacio será representado por el número 0. Por ejemplo, el estado mostrado en la figura 1 se representará con la lista: **[[1, 2, 3], [4, 5, 6], [7, 8, 0]]**.
- Un estado meta del 8-puzzle. Representado igual que el estado inicial (es decir, con una lista de listas de dígitos)
- El tipo de algoritmo a utilizar. Esta es una variable entera, si su valor es **0**, se debe usar BFS y si es **1**, DFS.

La función deberá regresar una lista que contenga la secuencia de acciones que deben realizarse para ir del estado inicial al estado final.

Ejemplo

	1	2
4	5	3
7	8	6

Estado Inicial

1	2	3
4	5	6
7	8	

Estado Final

Acciones:

R-R-D-D

La función se invocaría como:

```
edoInicial = [[0, 1, 2], [4, 5, 3], [7, 8, 6]]
edoFinal = [[1, 2, 3], [4, 5, 6], [7, 8, 0]]
busquedaNoInformada(edoInicial, edoFinal, 0) # puede llamarse con 1
```

Y deberá regresar una lista con los movimientos a realizar que logren resolver el problema, Por ejemplo:

[R, R, D, D]

Recuerde que hay un número infinito de soluciones para este problema. Pero también puede responder con una lista vacía, lo que indicaría que “NO HAY SOLUCION” para ese problema.

Consideraciones y entrega

1. Los movimientos se van a referir al espacio en blanco, el cual podrá moverse, si es posible, hacia arriba (U), hacia abajo (D), hacia la derecha (R) o hacia la izquierda (L).
2. El programa deberá resolver el problema usando los algoritmos de:
 - a. Búsqueda primero a lo ancho (*Breath-First Search*)
 - b. Búsqueda primero en profundidad (*Depth-First Search*)
3. Con el parámetro **algoritmo** el usuario deberá poder seleccionar por cuál de los dos algoritmos quiere obtener la solución.
4. **Se debe entregar todo lo indicado en el Syllabus.**

Recomendaciones

1. La estructura de datos usada para un nodo debe contener:
 - a. Estado: El estado al que corresponde el nodo, en este caso, un 8-puzzle
 - b. Papá: Apuntador a su nodo padre
 - c. Acción: La acción que se le aplicó al papá para llegar al nodo actualAlgunas veces conviene guardar el FRINGE completo, es decir, el conjunto de acciones que se han ejecutado para llegar desde el nodo raíz hasta el nodo actual.

Si se decide esto, es posible que no sea necesario guardar el nodo padre.

- d. Costo: El costo del camino desde el nodo raíz hasta el nodo actual

Probando el programa

- El programa será probado colocando todos los archivos necesarios para que corra en el mismo folder.
- El script utilizado (colocado en el archivo **main.py**) será el siguiente:

```
import busquedaNoInformada from Proyecto1.py
```

```
edoInicial = [listas que contengan el estado inicial]
```

```
edoFinal = [listas que contengan el estado final]
```

```
algoritmo = X      # donde X puede ser 0 para BFS o 1 para DFS
```

```
print(busquedaNoInformada(edoInicial, edoFinal, algoritmo))
```

- La función contestará con la lista de movimientos necesarios para ir del estado inicial al estado final o con una lista vacía en caso de que no haya solución.