

Sistemas Inteligentes

Proyecto 2

Búsquedas informadas: Algoritmo A*

Antecedentes

Uno de los principales algoritmos de búsqueda informada es el llamado A* propuesto por Peter Hart, Nils Nilsson and Bertram Raphael del Stanford Research Institute, en 1968.

El algoritmo trabaja con una función: $f(n) = g(n) + h(n)$

La función $f(n)$ toma en cuenta tanto el costo para llegar de la raíz al estado actual n ($g(n)$), como una estimación del costo para ir del estado actual n al estado meta ($h(n)$). Esta última estimación la hace por medio de una función heurística adecuada.

Heurísticas para el 8-puzzle

Para el 8-puzzle dos de las heurísticas más usadas son:

1. Número de fichas fuera de su lugar
2. Suma de distancias Manhattan de cada ficha a su posición correcta

Estas dos funciones heurísticas cumplen con las condiciones necesarias para que el algoritmo A* funcione correctamente.

Definición

Hacer una función en Python (el archivo que la contenga deberá llamarse **Proyecto2.py**) llamada:

busquedaAstar(edoInicial, edoFinal, heuristica)

que resuelva el problema del 8-puzzle utilizando el algoritmos A*. La función debe 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 heurística a utilizar. Esta es una variable entera, si su valor es **0**, se debe usar el número de cuadros fuera de su lugar y si es **1**, la suma de las distancias Manhattan de cada cuadro a su posición final.

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]]
búsquedaAstar(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

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 **A*** y permitirá al usuario seleccionar cualquiera de las dos heurísticas descritas anteriormente.
3. **Se debe entregar todo lo indicado en el Syllabus.**

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 Proyecto2.py
```

```
edoInicial = [listas que contengan el estado inicial]
edoFinal = [listas que contengan el estado final]
heuristica = X    # donde X puede ser 0 o 1
print(busquedaAstar(edoInicial, edoFinal, heuristica))
```

- 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.

Referencias

Una explicación completa del algoritmo A*, así como el algoritmo mismo lo pueden encontrar en:
https://en.wikipedia.org/wiki/A*_search_algorithm