



Tecnológico de Monterrey

Sistemas Inteligentes

Búsqueda Informada con A*

Manual de usuario

Profesor: Doctor Victor de la Cueva

Isaias Martinez Vieyra – A00988525

Manual de Usuario

Notas previas a correr el programa:

Es necesario tener instalado Python 3, la puedes descargar del siguiente link <https://www.python.org/downloads/> elige el link correspondiente al SO de tu maquina.

Una vez instalado Python 3 no necesitas hacer nada mas, para correr el programa que utilizaremos.

Si ya cuentas con Python3 es necesario verificar la version que tienes configurada en tus variables de ambiente para eso corremos el comando Python3 --version

```
MacBook-Pro-de-Marvi:~ marvi$ Python3 --version
Python 3.7.0
```

Pasos para correr el Programa:

1.Descargas el archivo Zip y extraes su contenido en la carpeta de tu elección.

2.Opción:

Puedes correr el archivo prueba.py el cual contiene dos matrices pre establecidas, “matriz inicio” y “matriz objetivo” puedes modificarlas y correr nuevamente el programa para ver la variación de pasos necesarios para llegar a la “matriz objetivo”.

También puedes modificar el tercer parámetro el cual define la Heurística a ocupar.

- a. 0 = Fuera de lugar
- b. 1 = Manhattan

Una vez corrido el programa prueba.py, veras en terminal impreso los pasos movimientos necesarios para que la “matriz inicio”, llegue a ser como la “matriz objetivo”.

a.Nomenclatura de los Movimientos:

- i.U = UP
- ii.D = DOWN
- iii.L = LEFT
- iv.R = RIGHT

Imagen de terminal al correr el archivo prueba.py y el resultado impreso

```
MacBook-Pro-de-Marvi:EntregaSegundoParcial marvi$ python3 prueba.py  
['U', 'R', 'R', 'D', 'D']
```

i. Opción:

Puedes importar el método a un nuevo documento de python con la siguiente línea “`from Proyecto2 import busquedaAstar`” y ocupar el método `busquedaAstar(p1,p2,p3)` el cual recibe tres parámetros, dos matrices y un número entero

- `p1` = matriz inicio
- `p2` = matriz objetivo.
- `P3` = Número entero 0 o 1

```
from Proyecto2 import busquedaAstar  
print(busquedaAstar([[4, 1, 2], [0, 5, 3], [7, 0, 6]], [[1, 2, 3], [4, 5, 6], [7, 0, 0]], 1))
```

i. Posibles resultados:

- b. Si la solución es encontrada, se imprimen los pasos necesarios para llegar a la matriz objetivo
- c. Matriz vacía, esto quiere decir que no se encontró una solución

Descripción del Código:

Dentro del Zip vienen 2 archivos `prueba.py` y `Proyecto2.py`.

En el archivo `Proyecto2`, contiene la clase `Nodo` la cual recibe 4 parámetros y una variable de manejo de heurística.

```
class Nodo:  
    def __init__(self, estado, padre, accion, profundidad):  
        self.estado = estado  
        self.padre = padre  
        self.accion = accion  
        self.profundidad = profundidad  
        self.heuristica = -1
```

- estado = Lista de listas
- padre = objeto tipo Nodo
- accion = Lista de Strings
- profundidad = Entero
- heurística = Entero

El archivo Proyecto2.py contiene una serie de métodos que ayudan a saber los pasos necesarios para llegar a la matriz objetivo, sin embargo el método principal es el método `busquedaAstar(p1,p2,p3)`, el cual recibe tres parámetros, dos matrices y un numero entero.

- p1 = matriz inicio
- p2 = matriz objetivo
- P3 = Numero entero 0 o 1

```
# 1 = Manhattan  0 = Cuadros fuera de lugar
def busquedaAstar(edoInicial, edoFinal, heuristica):

    inicial_list = [num for elem in edoInicial for num in elem]
    objetivo_list = [num for elem in edoFinal for num in elem]

    if heuristica == 0:
        return a_star(inicial_list, objetivo_list, fuera_lugar)
    elif heuristica == 1:
        return a_star(inicial_list, objetivo_list, manhattan)
    else:
        return []
```

El archivo prueba.py importa el método `busquedaAstar`, del archivo `Proyecto2.py` Y mandamos a llamar el método con dos matrices y un numero.

```
from Proyecto2 import busquedaAstar
print(busquedaAstar([[4, 1, 2], [0, 5, 3], [7, 8, 6]], [[1, 2, 3], [4, 5, 6], [7, 8, 0]], 1))
```