ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA
Universitat Rovira i Virgili

PRÁCTICA 1

Busqueda Informada

Alumno/s: Dickinson Bedoya Pérez
Asignatura: Inteligencia Artificial
Grau: Ingeniería Informática
Professor: David Sánchez
Fecha: 10/2022

ÍNDICE

Descripción del código **2**
Estructura de clases 2

Mapas **3**
Mapas 3

Explicación de los algoritmos **4**
Best First 4
A* 4

Resultados por cada mapa **5**
Mapa 1 Heurística 1 5
Mapa 1 Heurística 2 5
Mapa 1 Heurística 3 6
Mapa 2 Heurística 1 6
Mapa 2 Heurística 2 7
Mapa 2 Heurística 3 7
Mapa 3 8

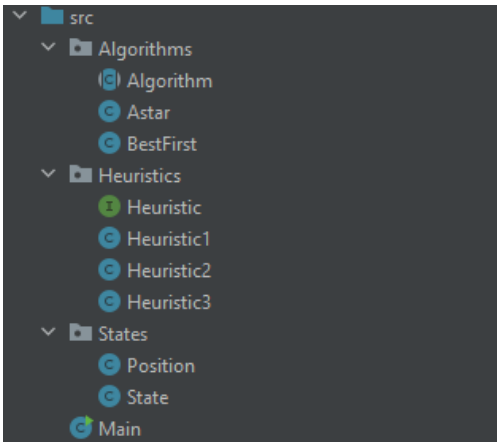
Resultados **8**

Descripción del código

Estructura de clases

Las clases e interfaces creadas en este proyecto tienen como fin tener una ejecución general e intuitiva del problema.
Se dividen en tres distintos paquetes:

1. Algorithm: Contiene la clase abstracta de la que heredan los dos algoritmos de búsqueda. Además tiene implementadas ciertas funciones que los dos algoritmos usan.
    a. Astar → Implementación del Algoritmo A*.
    b. BestFirst → Implementación del algoritmo Best First.
2. Heuristics: Contiene la interfaz de la que heredan todas las heurísticas. Se definen funciones que se implementan diferente por cada una de las heurísticas.
    a. Heuristic1: Implementación de la heurística 1. Esta heurística se basa en comparar los pesos de los estados para decidir cuál será el siguiente candidato.
    b. Heuristic2: Implementación de la heurística 2. Se basa en comparar las coordenadas de cada estado con el estado final para hacer una especie de distancia. A esta distancia se le suma la altura de cada estado y así se puede ver con cual quedarse.
    c. Heuristic3: Implementación de la heurística 3. Basado en la distancia euclidiana de dos puntos, se le añade la altura para hacer una especie de heurística 2 pero mas completa.
3. States: Contiene las clases que definen cada una de las casillas del mapa.
    a. Position: Posición del estado en el mapa a partir de su fila y columna.
    b. State: Definición de cada estado. Tiene su posición en el mapa y su altura.



Mapas

Se han probado 3 mapas, cada uno con una distribución diferente de las alturas y barrancos. Los dos primeros (uno de ellos el dado en clase) tienen solución y el último no la tiene. En todos el objetivo es ir desde la casilla 0,0 hasta la 9,9.

Mapas

```
MAPA BASE
|1 0 X 1 3 2 3 4 3 1|
|2 1 X 2 4 2 2 4 2 2|
|5 3 X 2 3 2 X 3 3 3|
|3 3 1 3 4 3 X 1 2 2|
|2 2 2 3 6 4 X 1 2 1|
|X X X X 3 3 X 1 2 X|
|X X X X 2 4 X 2 2 X|
|2 3 4 3 1 3 X 3 2 X|
|3 5 6 5 2 3 X 5 3 X|
|5 6 7 6 4 4 X 6 4 5|
```

Mapa proporcionado en clase.

```
MAPA BASE
|1 0 1 1 3 2 X 4 3 1|
|2 1 2 2 4 2 X 4 2 2|
|X X X 2 3 X 3 3 3 4|
|3 3 1 3 4 3 X 1 2 2|
|2 2 2 3 6 4 X 1 2 1|
|X X X X 3 3 X 1 2 X|
|X X X X 2 4 X 2 2 X|
|2 3 4 3 1 3 1 3 2 X|
|3 5 6 5 2 3 4 5 3 X|
|5 6 7 6 4 4 5 6 4 5|
```

Mapa con solución.

```
MAPA BASE
|1 0 X 1 3 2 3 4 3 1|
|2 1 X 2 4 2 X X X X|
|5 3 2 2 3 2 X 3 3 3|
|X X X 3 4 3 X 1 2 2|
|2 2 2 3 6 4 X 1 2 1|
|X X X X 3 3 X 1 2 X|
|X X X X 2 4 X 2 2 X|
|2 3 4 3 1 3 X 3 2 X|
|3 5 6 5 2 3 X 5 3 X|
|5 6 7 6 4 4 X 6 4 5|
```

Mapa sin solución

Explicación de los algoritmos

Best First

Este algoritmo, como dice su nombre, coge el primer mejor camino, no importa si es el más óptimo.
Con esto, se consigue que no haya tanto tratado de estados y no haya tanta memoria ocupada.
La elección de cuál será el siguiente estado a tratar viene dada directamente de la heurística.

A*

Este algoritmo llega a tratar todas las posibilidades y escoge el mejor camino.

La elección del siguiente estado a tratar, a diferencia del Best First, viene dado por una fórmula:

$$f(x) = g(x) + h(x)$$

Donde $g(x)$ es el coste que hay para llegar a ese nodo. En mi caso, es la suma de alturas que hay para llegar hasta ese nodo.
Si la fórmula fuese $f(x) = g(x)$ sería un algoritmo Dijkstra normal.

Resultados por cada mapa

Mapa 1 Heurística 1

```
Algoritmo Best First con Heuristica N°1
Numero de nodos tratados: 63
Coste: 29.0
Camino:
|1  0  X  1  3  2  3  4  3  1|
|2  1  X  2  4  2  2  4  2  2|
|5  3  X  2  3  2  X  3  3  3|
|3  3  1  3  4  3  X  1  2  2|
|2  2  2  3  6  4  X  1  2  1|
|X  X  X  X  3  3  X  1  2  X|
|X  X  X  X  2  4  X  2  2  X|
|2  3  4  3  1  3  X  3  2  X|
|3  5  6  5  2  3  X  5  3  X|
|5  6  7  6  4  4  X  6  4  5|
```

```
Algoritmo A* con Heuristica N°1
Numero de nodos tratados: 69
Coste: 29.5
Camino:
|1  0  X  1  3  2  3  4  3  1|
|2  1  X  2  4  2  2  4  2  2|
|5  3  X  2  3  2  X  3  3  3|
|3  3  1  3  4  3  X  1  2  2|
|2  2  2  3  6  4  X  1  2  1|
|X  X  X  X  3  3  X  1  2  X|
|X  X  X  X  2  4  X  2  2  X|
|2  3  4  3  1  3  X  3  2  X|
|3  5  6  5  2  3  X  5  3  X|
|5  6  7  6  4  4  X  6  4  5|
```

Mapa 1 Heurística 2

```
Algoritmo Best First con Heuristica N°2
Numero de nodos tratados: 49
Coste: 31.0
Camino:
|1  0  X  1  3  2  3  4  3  1|
|2  1  X  2  4  2  2  4  2  2|
|5  3  X  2  3  2  X  3  3  3|
|3  3  1  3  4  3  X  1  2  2|
|2  2  2  3  6  4  X  1  2  1|
|X  X  X  X  3  3  X  1  2  X|
|X  X  X  X  2  4  X  2  2  X|
|2  3  4  3  1  3  X  3  2  X|
|3  5  6  5  2  3  X  5  3  X|
|5  6  7  6  4  4  X  6  4  5|
```

```
Algoritmo A* con Heuristica N°2
Numero de nodos tratados: 65
Coste: 29.5
Camino:
|1  0  X  1  3  2  3  4  3  1|
|2  1  X  2  4  2  2  4  2  2|
|5  3  X  2  3  2  X  3  3  3|
|3  3  1  3  4  3  X  1  2  2|
|2  2  2  3  6  4  X  1  2  1|
|X  X  X  X  3  3  X  1  2  X|
|X  X  X  X  2  4  X  2  2  X|
|2  3  4  3  1  3  X  3  2  X|
|3  5  6  5  2  3  X  5  3  X|
|5  6  7  6  4  4  X  6  4  5|
```

Mapa 1 Heurística 3

```
Algoritmo Best First con Heuristica N°3
Numero de nodos tratados: 49
Coste: 31.0
Camino:
|1  0  X  1  3  2  3  4  3  1|
|2  1  X  2  4  2  2  4  2  2|
|5  3  X  2  3  2  X  3  3  3|
|3  3  1  3  4  3  X  1  2  2|
|2  2  2  3  6  4  X  1  2  1|
|X  X  X  X  3  3  X  1  2  X|
|X  X  X  X  2  4  X  2  2  X|
|2  3  4  3  1  3  X  3  2  X|
|3  5  6  5  2  3  X  5  3  X|
|5  6  7  6  4  4  X  6  4  5|
```

```
Algoritmo A* con Heuristica N°3
Numero de nodos tratados: 76
Coste: 33.0
Camino:
|1  0  X  1  3  2  3  4  3  1|
|2  1  X  2  4  2  2  4  2  2|
|5  3  X  2  3  2  X  3  3  3|
|3  3  1  3  4  3  X  1  2  2|
|2  2  2  3  6  4  X  1  2  1|
|X  X  X  X  3  3  X  1  2  X|
|X  X  X  X  2  4  X  2  2  X|
|2  3  4  3  1  3  X  3  2  X|
|3  5  6  5  2  3  X  5  3  X|
|5  6  7  6  4  4  X  6  4  5|
```

Mapa 2 Heurística 1

```
Algoritmo Best First con Heuristica N°1
Numero de nodos tratados: 57
Coste: 24.0
Camino:
|1  0  1  1  3  2  X  4  3  1|
|2  1  2  2  4  2  X  4  2  2|
|X  X  X  2  3  X  3  3  3  4|
|3  3  1  3  4  3  X  1  2  2|
|2  2  2  3  6  4  X  1  2  1|
|X  X  X  X  3  3  X  1  2  X|
|X  X  X  X  2  4  X  2  2  X|
|2  3  4  3  1  3  1  3  2  X|
|3  5  6  5  2  3  4  5  3  X|
|5  6  7  6  4  4  5  6  4  5|
```

```
Algoritmo A* con Heuristica N°1
Numero de nodos tratados: 58
Coste: 26.5
Camino:
|1  0  1  1  3  2  X  4  3  1|
|2  1  2  2  4  2  X  4  2  2|
|X  X  X  2  3  X  3  3  3  4|
|3  3  1  3  4  3  X  1  2  2|
|2  2  2  3  6  4  X  1  2  1|
|X  X  X  X  3  3  X  1  2  X|
|X  X  X  X  2  4  X  2  2  X|
|2  3  4  3  1  3  1  3  2  X|
|3  5  6  5  2  3  4  5  3  X|
|5  6  7  6  4  4  5  6  4  5|
```

Mapa 2 Heurística 2

```
Algoritmo Best First con Heuristica N°2
Numero de nodos tratados: 20
Coste: 24.0
Camino:
|1  0  1  1  3  2  X  4  3  1|
|2  1  2  2  4  2  X  4  2  2|
|X  X  X  2  3  X  3  3  3  4|
|3  3  1  3  4  3  X  1  2  2|
|2  2  2  3  6  4  X  1  2  1|
|X  X  X  X  3  3  X  1  2  X|
|X  X  X  X  2  4  X  2  2  X|
|2  3  4  3  1  3  1  3  2  X|
|3  5  6  5  2  3  4  5  3  X|
|5  6  7  6  4  4  5  6  4  5|
```

```
Algoritmo A* con Heuristica N°2
Numero de nodos tratados: 44
Coste: 29.0
Camino:
|1  0  1  1  3  2  X  4  3  1|
|2  1  2  2  4  2  X  4  2  2|
|X  X  X  2  3  X  3  3  3  4|
|3  3  1  3  4  3  X  1  2  2|
|2  2  2  3  6  4  X  1  2  1|
|X  X  X  X  3  3  X  1  2  X|
|X  X  X  X  2  4  X  2  2  X|
|2  3  4  3  1  3  1  3  2  X|
|3  5  6  5  2  3  4  5  3  X|
|5  6  7  6  4  4  5  6  4  5|
```

Mapa 2 Heurística 3

```
Algoritmo Best First con Heuristica N°3
Numero de nodos tratados: 21
Coste: 24.0
Camino:
|1  0  1  1  3  2  X  4  3  1|
|2  1  2  2  4  2  X  4  2  2|
|X  X  X  2  3  X  3  3  3  4|
|3  3  1  3  4  3  X  1  2  2|
|2  2  2  3  6  4  X  1  2  1|
|X  X  X  X  3  3  X  1  2  X|
|X  X  X  X  2  4  X  2  2  X|
|2  3  4  3  1  3  1  3  2  X|
|3  5  6  5  2  3  4  5  3  X|
|5  6  7  6  4  4  5  6  4  5|
```

```
Algoritmo A* con Heuristica N°3
Numero de nodos tratados: 58
Coste: 29.0
Camino:
|1  0  1  1  3  2  X  4  3  1|
|2  1  2  2  4  2  X  4  2  2|
|X  X  X  2  3  X  3  3  3  4|
|3  3  1  3  4  3  X  1  2  2|
|2  2  2  3  6  4  X  1  2  1|
|X  X  X  X  3  3  X  1  2  X|
|X  X  X  X  2  4  X  2  2  X|
|2  3  4  3  1  3  1  3  2  X|
|3  5  6  5  2  3  4  5  3  X|
|5  6  7  6  4  4  5  6  4  5|
```

Mapa 3

```
Algoritmo Best First con Heuristica N°1
Numero de nodos tratados: 49
Coste: 0.0
No se ha podidio encontrar el camino


Algoritmo Best First con Heuristica N°2
Numero de nodos tratados: 49
Coste: 0.0
No se ha podidio encontrar el camino


Algoritmo Best First con Heuristica N°3
Numero de nodos tratados: 49
Coste: 0.0
No se ha podidio encontrar el camino
```

```
Algoritmo A* con Heuristica N°1
Numero de nodos tratados: 49
Coste: 0.0
No se ha podidio encontrar el camino


Algoritmo A* con Heuristica N°2
Numero de nodos tratados: 49
Coste: 0.0
No se ha podidio encontrar el camino


Algoritmo A* con Heuristica N°3
Numero de nodos tratados: 49
Coste: 0.0
No se ha podidio encontrar el camino
```

Resultados

Como se ve en los resultados, directamente para el mapa 3 ninguno de los dos algoritmos con sus heurísticas puede encontrar el camino, que es correcto.
Para los mapas 1 y 2 los resultados son diferentes. Los costes de tiempo van entre los 24 y los 33 como mucho. Aún y así, en ciertos casos se ve que el algoritmo A* encuentra un coste temporal menor y en otros el Best First.
Que el Best First tenga mejores resultados que el A* no implica que el A* sea peor algoritmo, sino que, al no tener las mejores heurísticas no encuentra el mejor camino.
También, en los resultados se puede ver que el algoritmo A* llega a tratar un número mayor de nodos, cosa que puede suponer una pequeña ventaja en encontrar el mejor camino.

La heurísticas 2 y 3 pueden encontrar solución con el algoritmo ***Hill Climbing*** ya que este algoritmo coge los mejores sucesores si son mejores que el padre ya que cada sucesor está más cerca del estado final.

Con la heurística 1 no puedo decir lo mismo ya que puede ser que el algoritmo se quede estancado al encontrarse con que los estados sucesores tengan una altura mayor que la suya y no continúe.