

## Entrega 4

Jon Acha 79001780T

Sergio Romero 78993882Z

*Pregunta 1: Suponiendo que sólo es posible desplazarse horizontal o verticalmente casilla a casilla, considera las siguientes medidas de:*

- *Distancia en línea recta (Distancia Euclídea). Esta es la misma distancia que usaste en la Parte 1 ya que en ese caso había un camino directo entre cada par de objetos. Ahora debes determinar si esa medida de distancia es una estimación admisible de la distancia real entre cada par de objetos teniendo en cuenta el tipo de movimiento descrito.*
- *Distancia de Manhattan.*
- *Distancia de Chebyshev.*

*¿Cuáles son admisibles para estimar la distancia entre 2 objetos? ¿Cuál utilizarías y por qué? (JUSTIFICA TUS RESPUESTAS).*

La distancia Euclídea calcula la distancia en línea recta hacia la distancia más corta. Por lo tanto para nuestro problema como no hay caminos directos no es admisible.

La distancia de Manhattan es admisible pues calcula la distancia no en línea recta, sino en movimientos verticales y horizontales o diagonales respectivamente. La distancia de Manhattan realizara el camino con las variaciones en el recorrido entre dos puntos de arriba-abajo e izquierda-derecha.

La distancia Chebyshev también no es admisible, porque no solo se mueve de la misma forma que la distancia Manhattan sino que también de forma diagonal.

Utilizaremos la distancia Manhattan ya que puede desplazarse entre dos puntos por medio de movimientos verticales y horizontales.

*Pregunta 2: Si además del movimiento en vertical y horizontal, es posible el desplazamiento en diagonal casilla a casilla: ¿Cuáles de las 3 medidas de distancia siguen siendo admisibles para estimar la distancia entre 2 objetos?*

La distancia Euclídea no es admisible por la misma razón que en la anterior pregunta.

Manhattan no es admisible, pues no realiza movimientos diagonales.

Chebyshev será admisible ya que es la única que puede calcular movimientos en diagonal lo que reduciría el coste de la distancia.

*Pregunta 3: Coste Uniforme es un caso especial de otro algoritmo de búsqueda más general. ¿Cuál es ese algoritmo? Explica cómo (bajo qué condiciones) ese algoritmo se convierte en la búsqueda de Coste Uniforme.*

Ese algoritmo más general es el Breadth First. Este algoritmo se convierte en coste uniforme cuando almacenamos todas las acciones posibles en una lista FIFO ordenada según el coste, se elige la acción de menor coste, en caso de que todas las acciones tengan el mismo coste, se aplica directamente Breadth First.

### *Análisis del rendimiento del sistema.*

La solución dada por el algoritmo de la distancia en línea recta y en la de manhattan es la más óptima. Siempre saldrá la misma solución sin importar la función de evaluación que utilicemos la solución es: S-A-B-C-D-E-G-F-Z.

Hemos decidido utilizar el algoritmo Tree-search, pues es el que devuelve un recorrido más óptimo, a diferencia de Graph-search que pese a generar menos estados repetidos no nos devuelve un camino óptimo. La pega de Tree-search es que realiza un gran número de estados repetidos.

El resultado devuelto es un recorrido total de 30.06465 en un tiempo de ejecución de 245 ms mediante el algoritmo de ordenación Manhattan. En este algoritmo de ordenación se va realizando cálculos adicionales cíclicos para realizar la ordenación requerida y por eso el tiempo de ejecución es mayor. El resultado devuelto por el algoritmo de la distancia en línea recta es el mismo, pero tarda 106 ms debido a que tenemos un atributo distanciaTotal en la clase Entorno que se va actualizando según nos desplazamos por las ciudades, por lo tanto en este caso no necesitamos realizar cálculo adicional.