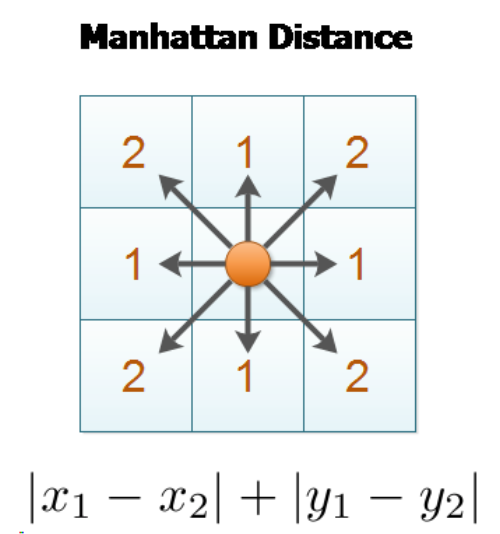
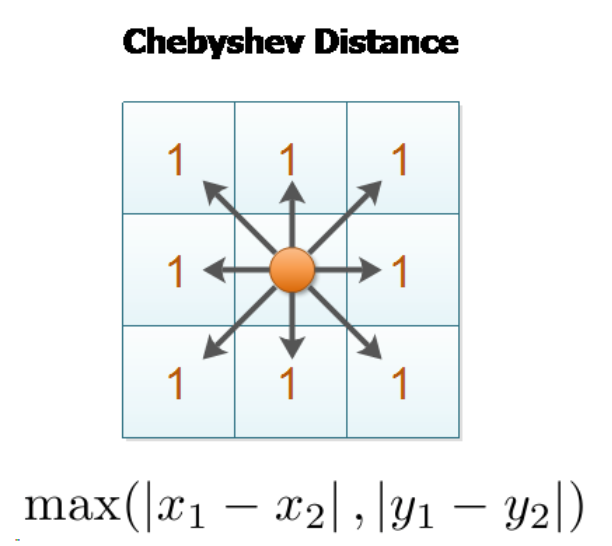
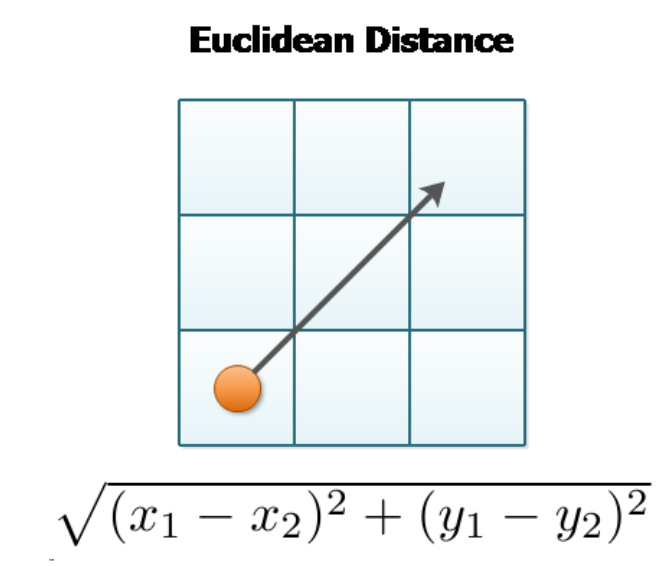
Homework 4

Preguntas

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

* Distancia en linea recta (Distancia Euclidea): 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

La mejor de todas sería Manhattan debido a lo siguiente:

Se establece la premisa de que solo nos podemos desplazar horizontal y verticalmente, luego para ir a una casilla que es diagonal el coste de movimiento será de 2 no de uno, así que la implementación más lógica sería la de Manhattan Distance.

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?

En este caso la mejor medida a implementar sería Chebyshev dado que el movimiento diagonal es posible el coste de moverse hacia otra en diagonal es de 1, no de dos por los dos movimientos verticales y horizontales.

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.

El algoritmo es A\*, que es un caso genérico del algoritmo de coste uniforme. Si para todo h el valor es 0 entonces el algoritmo A\* se convierte en coste uniforme, también si el valor de h siempre es el mismo.

Justificaciones

He elegido el Tree Search porque no se generan en este tipo de problema bucles dado que no puede volver a una ciudad en el que ya se a estado luego no tiene sentido implementar un algoritmo más costoso como Graph Search para controlar una circunstancia que no se va a dar.

Estudio de susceptibilidad a la ordenación de los operadores

Da igual como esten ordenados los operadores, el resultado siempre es el mismo

[S A B C D E G F Z]

Sin importar que tipo de coste se aplique.

Optimización

El algoritmo ya ha logrado resolver el problema con su coste óptimo en distancia, tal vez podría mejorarse más en tiempo pero no se me ocurre como optimizar ya más el tiempo de ejecución del programa de manera que se vea visible, es decir ya no alguna parte del código que no se ejecute de manera óptima y que pudiese ser optimizada de manera relevante.

Aunque si es cierto que dada la descripción del problema podríamos en isApplicable quitar la comprobación de si desde una ciudad podemos ir a otra dado que en la descripción del problema podemos ir desde cualquiera a cualquiera, esto he echo pruebas y reduce bastante el tiempo de ejecución, le ahorra aproximadamente unos 100ms pero prefiero dejarlo dado que expande la posibilidad de adaptar el problema a otras circunstancias.