

Universidad Autónoma de Ciudad Juárez

Proyecto de Búsqueda en Grafos

Sistemas Inteligentes

Dr. Nahitt Padilla

1.- Introducción

El metro de Londres consiste en 306 estaciones divididas en 13 líneas, ver mapa en la Figura 1.

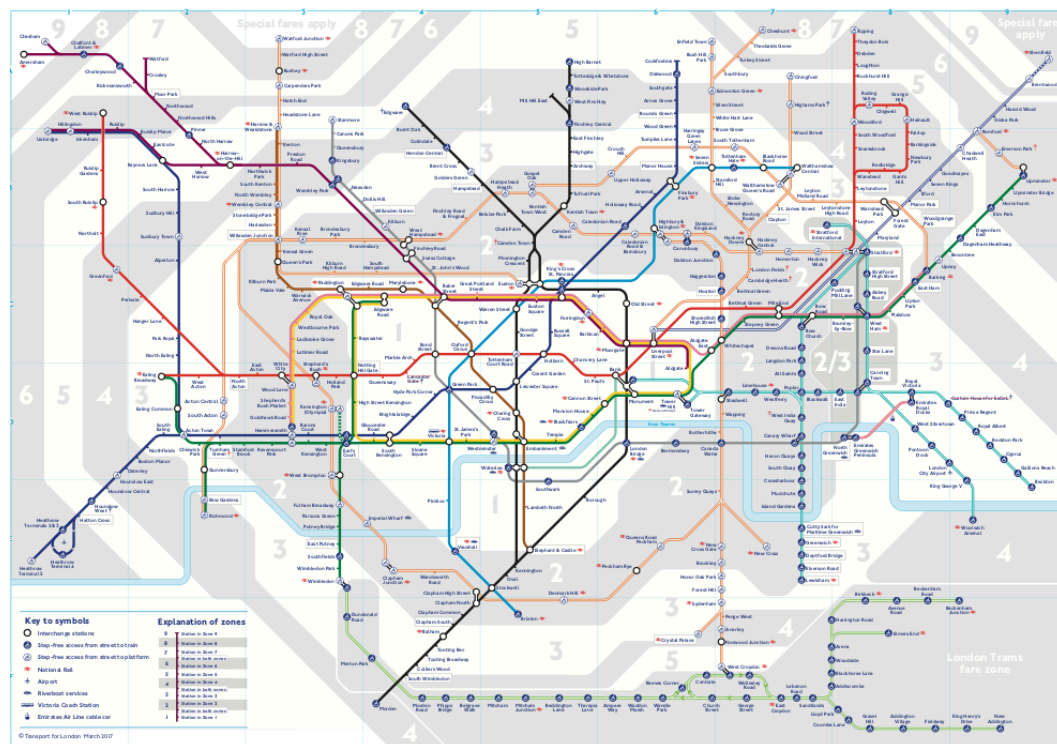


Figura 1. Metro de Londres

Existe una aplicación en internet que calcula las ruta entre dos puntos del metro de Londres <https://tfl.gov.uk/plan-a-journey/>, la interfaz de entrada de datos se puede ver en la Figura 2.

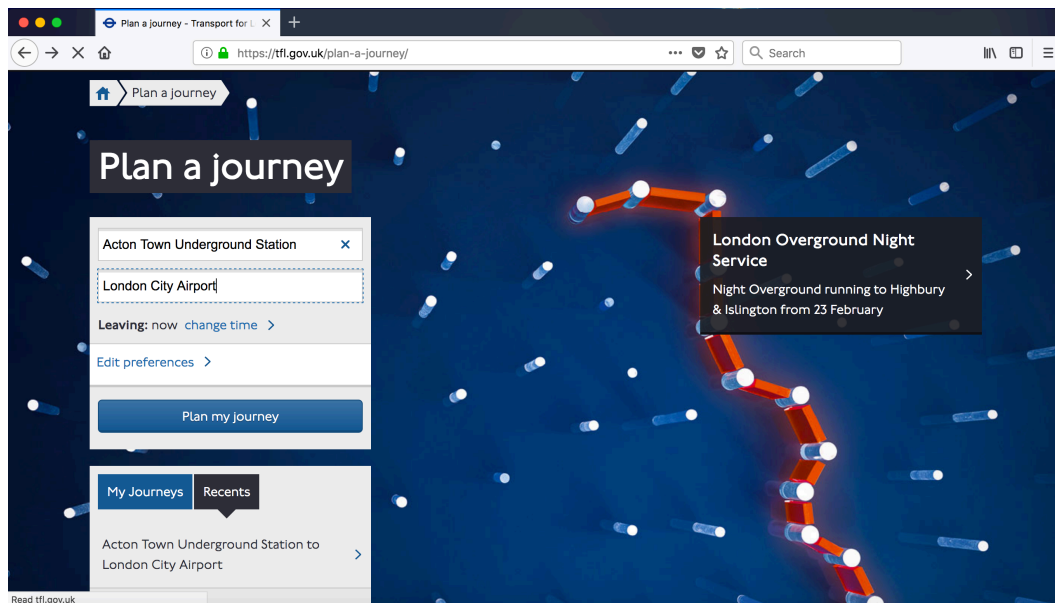


Figure 2. Plan a journey web site.

Después de ingresar las estaciones de salida y de destino se debe presionar el botón de planea mi viaje “Plan my journey” para obtener una ruta. El viaje resultado consiste en la lista de estaciones intermedias que el usuario debe de viajar para alcanzar su destino, véase la Figura 3.

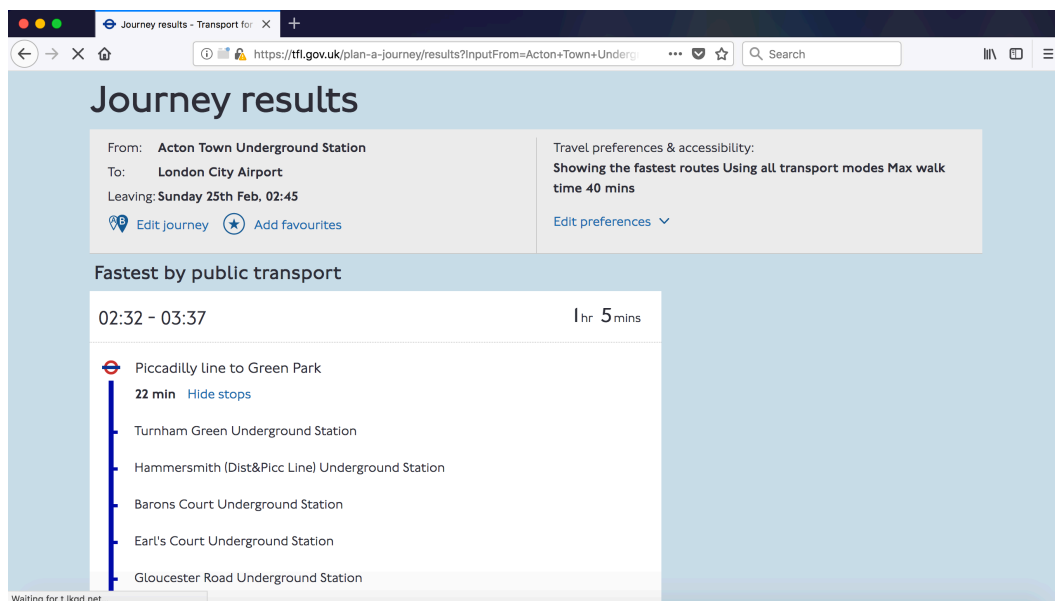


Figura 3. Resultado de la ruta buscada

2.- Objetivo

El objetivo de este proyecto es crear una aplicación que permita encontrar la ruta entre dos estaciones proporcionadas por el usuario. Para lograr eso es necesario representar mediante un grafo el sistema de transporte subterráneo de Londres. Un ejemplo de visualización de este grafo se puede observar en la Figura 4.



Figura 4. Grafo del Metro de Londres. Tomado de <https://visualign.wordpress.com/2012/07/11/london-tube-map-and-graph-visualizations/>

Una vez que se cuente con la representación de grafo, se puede implementar el algoritmo de búsqueda visto en clase para encontrar dicha ruta.

3.- Dataset

El metro de Londres es muy común encontrarlo como ejemplo de grafos debido a que su Dataset es gratis. Se puede descargar en https://commons.wikimedia.org/wiki/London_Underground_geographic_maps/CSV

El dataset consiste en tres archivos: stations.csv, lines.csv y lineDefinition.csv

El primer archivo stations.csv define la posición de la estación, su nombre y su clave. Este archivo tiene los siguientes campos separados por comas:

"id","latitude","longitude","name","display_name","zone","total_lines","rail"

Donde “id” representa la clave de la estación. Longitud y latitud su ubicación geográfica. Name es una cadena que describe el nombre de la estación. Display_name es el nombre en html, y se utiliza para mostrar el nombre de la estación a los usuarios. Zone define a que zona pertenece la estación. Total_lines es el número de líneas que llegan a una estación. Y finalmente rail dice si la estación del metro se conecta con la estación de ferrocarril. Por ejemplo, el primer registro del archivo es

1,51.5028,-0.2801,"Acton Town","Acton
Town",3,2,0

Que se interpreta que la estación clave 1, se encuentra en la latitud 51.5028 y el la longitud -0.2801. El nombre de la estación es Acton Town y se mostrara a los usuarios con una separación de línea entre ambas palabras (
). Esta estación está en la zona 3 y conecta dos líneas del metro. El ultimo cero indica que no tiene conexión con el ferrocarril.

El siguiente archivo lines.csv tiene el siguiente formato:

"line","name","colour","stripe"

Donde line es el número de línea. Name es el nombre de la línea. Colour es el código de color de la línea en RGB y stripe es si el color de la línea tiene una raya de otro color. Por ejemplo, el primer registro del archivo indica lo siguiente:

1,"Bakerloo Line","ab6612",NULL

Que se interpreta como que la línea 1 se llama “Bakerloo Line” y tiene un color en hexadecimal ab6612 que no tiene raya en medio.

El ultimo archivo es lineDefintion.csv y tiene un formato de

"station1","station2","line"

Donde Station 1 es la estación origen y station 2 es la estación destino, indicando que hay una conexión bidireccional entre las dos estaciones. Esta conexión pertenece a la línea descrita por el campo line. Como ejemplo el primer registro indica que

11,163,1

Que la estación 11 está conectada con la estación 163 y que esta conexión pertenece a la línea 1.

4.- Implementación

El proyecto consiste en dos partes: el preprocesamiento y la búsqueda. El preprocesamiento consiste en abrir los archivos del dataset y transformarlos automáticamente a una representación que permita su utilización por el algoritmo de búsqueda. En otras palabras, un programa que lea

los archivos separados por comas (CSV) y lo represente en una matriz de incidencia en algún lenguaje de programación.

La parte de la búsqueda implica la implementación del algoritmo de búsqueda expuesto en clase con sus dos estrategias de búsqueda: en anchura y en profundidad.

5.- Productos a entregar

Un programa que transforme los archivos CSV a una estructura de datos que pueda ser usada en un lenguaje de programación.

Un programa que contenga el grafo generado por el programa anterior y que realice la búsqueda en anchura y en profundidad.

Para lograr esto el programa le preguntara al usuario un nombre de estación de origen y un nombre de estación destino. Después le pedirá que tipo de búsqueda debe de realizar (anchura o profundidad)

El resultado se presentará como una serie de nombres de estaciones con su nombre de línea. Ambos programas pueden ser implementados en cualquier lenguaje de programación, pudiendo inclusive ser destinos para ambos programas, e.g., el preprocesamiento en Python, la búsqueda en Java.