

TravelMap GT

La empresa Code&Bugs ha analizado su dos prácticas anteriores, dadas sus habilidades ha decidido confiar en usted, por lo que se le contrata para el proyecto TravelMapGT, el cual es una aplicación basada en algunas características de Google Maps. Por lo cual el sistema deberá permitir a las personas de Guatemala organizar sus viajes desde un punto A a un punto B por medio de las siguientes opciones.

- Vehículo
- Caminando

La aplicación consiste en presentar sugerencias de las rutas más idóneas para poder llegar a su destino. Para esto se debe considerar el tiempo de recorrido (tomando en cuenta los horarios del tráfico) y la eficiencia. Para lo cual es necesario tomar en cuenta los siguientes elementos.

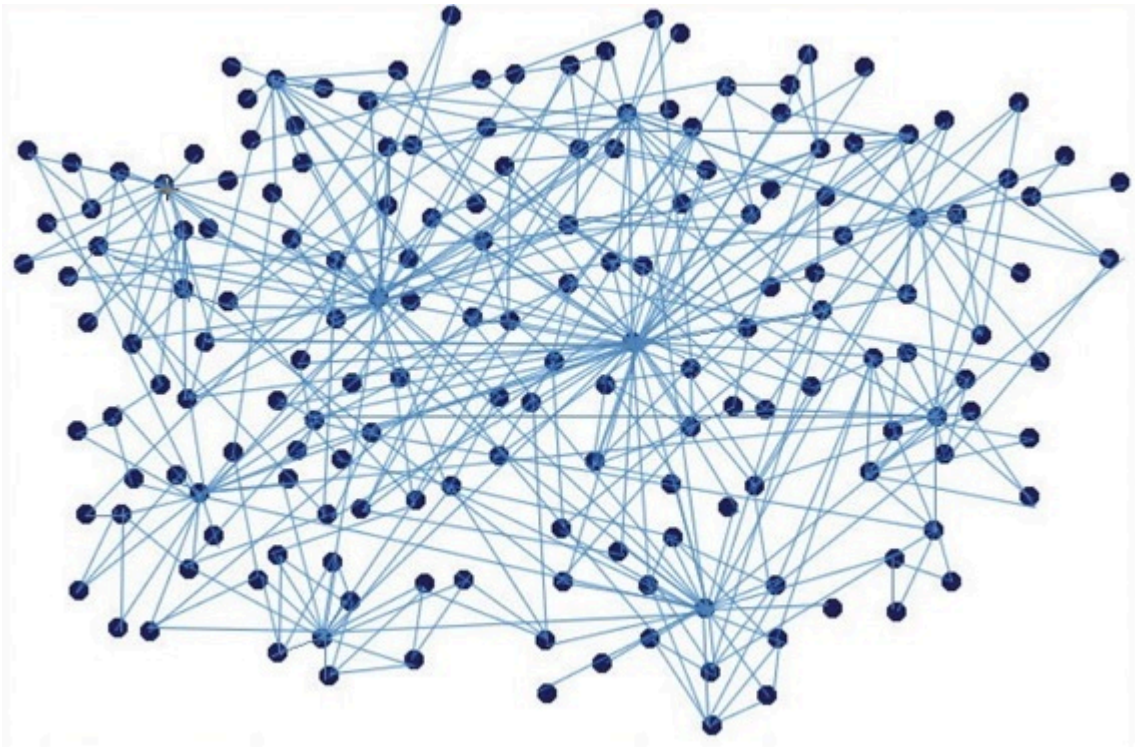
- Distancia: Tomar en cuenta la gasolina o esfuerzo físico.
- Rapidez de la ruta.

Es importante mostrar las demás rutas posibles cercanas. El usuario puede decidir que ruta tomar. También deberá tomar en cuenta que si el usuario toma otra ruta la cual no es más eficiente y se mueve de un punto a otro deberá de re-calcular en cada paso (entre nodos) que da la persona ,la ruta más eficiente y sus posibles rutas de llegada. Para esta aplicación deberá de plantear el tipo de estructura que se le pide a continuación para que de esta forma el cálculo de la ruta sea la más eficiente en cuanto al desarrollo se refiere.



Estructura de Datos

Para poder realizar la siguiente aplicación usted deberá de apoyarse de estructuras de datos la primera que deberá de tomar en cuenta son los grafos, como el que se ve en la siguiente imagen, se representa el país de Guatemala cada uno de los nodos es un posible destino al que nosotros podemos llegar y cada una de las líneas es una ruta que conecta los puntos. Debe ser un grafo dirigido, debe considerar si la ruta tiene doble vía, una vía o no, para los casos en que el usuario vaya en vehículo.



Para poder realizar dicho grafo usted tendrá que aceptar un archivo de entrada el cual contendrá la información de las posibles rutas y los diferentes destinos el cual tendrá la siguiente estructura:

<origen>|<destino>|<tiempo_vehiculo>|<tiempo_pie>|<consumo_gas>|<desgaste_persona>|<distancia>

en donde:

- origen es la ubicación de la partida
- destino es la ubicación de llegada
- El tercer dato es el tiempo promedio de llegada en vehículo
- El cuarto es el tiempo aproximado si una persona va caminando
- El quinto es el gasto en combustible de un vehículo
- El sexto es el gasto físico que tendrá la persona en este recorrido.
- El último es la distancia que existe entre el origen y el destino

Ejemplo:

```
AltaVerapaz|SantaMaríadeJesús|24|51|24|456|85
Petén|CiudadVieja|24|42|24|453|24
Quetzaltenango|Retalhuleu|25|50|56|54|100
Retalhuleu|Quetzaltenango|36|45|60|80|100
SanMarcos|Quetzaltenango|30|35|62|15|250
Suchitepequez|Guatemala|70|250|60|2500|70
SanMarcos|SanBartoloméMilpasAltas|24|425|62|63|55
Huehuetenango|SanLucasSacatepéquez|24|123|32|47|62
Quiché|SantaLucíaMilpasAltas|24|62|42|46|32
```

Todos los valores numéricos serán enteros.

Para los datos de los horarios de trafico, se cargara desde un segundo archivo de entrada donde contendrá la siguiente información sobre el tráfico de los distintos destinos, la estructura es la siguiente:

```
<origen>|<destino>|<hora_inicio>|<hora_finaliza>|<probabilidad_trafico>
```

en donde:

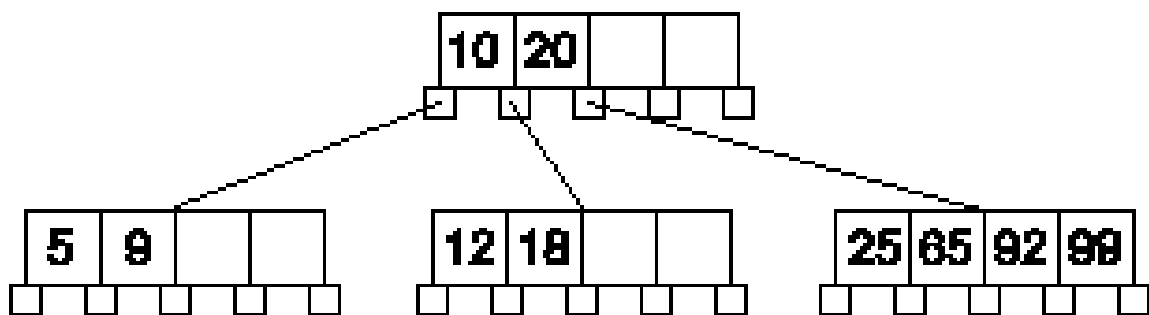
- origen es la ubicación de la partida
- destino es la ubicación de llegada
- El tercero es la hora en la que inicia el tráfico en esa ruta
- El cuarto es la hora en la que finaliza el tráfico en esa ruta, y vuelve a su normalidad (probabilidad_trafico = 0).
- El último es la probabilidad de tráfico que podría existir en esa ruta en ese horario.

Ejemplo:

```
Quetzaltenango|Retalhuleu|12|14|40  
Retalhuleu|Quetzaltenango|5|8|45  
Retalhuleu|Quetzaltenango|15|18|80  
Suchitepéquez|Retalhuleu|13|15|40  
Quetzaltenango|Salola|12|14|40  
Retalhuleu|Suchitepequez|16|21|63
```

Todos los valores numéricos serán enteros. Los horarios serán tomados con formato 24H, es decir que si tiene 15 será igual a decir 3:00PM. Para la probabilidad de tráfico los enteros podrán ser de 0 - 100, por lo tanto si tiene un 80 es igual a decir 80% o 0.80 de probabilidad de trafico.

Cómo se habló anteriormente cuando una persona se mueva de un nodo hacia otro, la aplicación deberá recalcular las rutas posibles para poder llegar a su destino, el usuario podrá marcar en qué punto actual se encuentra. Para recalcular las nuevas rutas posibles, cada uno de los nodos que conforman una ruta posible, deberán ser almacenados en una lista, esta debe de tener un identificador para ser almacenada dentro de un árbol tipo B.



El árbol se vería de esta forma, y cada uno de los números que aparecen dentro del árbol corresponde al id de una lista de nodos de una posible ruta para poder llegar al destino. Para la realización de este árbol toma en cuenta que el grado del árbol será de $m = 5$.

Sin embargo el id de la lista no se deberá demostrar como en la gráfica de arriba, el id únicamente servirá para insertar la lista pero lo que en realidad se tiene que mostrar son los sitios por los cuales pasará en dicha ruta. Para dibujar este tipo de árbol como también el grafo del mapa tendrá que apoyarse de la herramienta graphviz.

Algo muy importante que debe de tener en cuenta es que el usuario puede ir a pie o puede ir en vehículo, pero si la opción es ir en vehículo deberá de tomar en cuenta

si hay vía para seleccionar las rutas si por el contrario el usuario va caminando, no será necesario verificar si hay doble vía o no.

Funcionalidad del Tráfico

Dado que el sistema cuenta con datos sobre el posible tráfico que existe entre rutas, este usara un reloj, en el que deberá poder mostrar la hora actual, pero también este reloj debe ser editable para colocar cualquier hora y tambien se podra bloquear la hora para que esta no pueda seguir avanzando, de manera que la hora que se encuentre el reloj se usará para determinar los resultados obtenidos en las rutas, es decir que si la hora en la que se encuentra esta dentro de los horarios de tráfico de la ruta, se usará el dato de probabilidad tráfico para obtener los resultados en base a esta información adicional, por lo tanto este dato hará que todo el sistemas de sugerencia de rutas pueda cambiar sus resultados.

Tome en cuenta que para los cálculos deber de sumar la hora en la que se encuentra y el tiempo del punto A al punto B, y así sucesivamente según la ruta.

Cálculo de Datos:

Rapidez (Vehículo) = distancia / (tiempo x (1 + probabilidad_trafico))

Rapidez (Caminando) = (distancia / tiempo)

Funcionalidad

- La mejor ruta en base a la gasolina si es vehículo.
- La mejor ruta en base al desgaste físico si es caminando.
- La mejor ruta en base a la distancia.
- La mejor ruta en base a la gasolina y la distancia si es en vehículo.
- La mejor ruta en base al desgaste físico y la distancia si es caminando.
- La ruta más rápida en base a la distancia, tiempo y probabilidad de tráfico (tráfico solo si va en vehículo), para cualquiera de los dos tipos de transporte.
- Las peores rutas en los anteriores casos.

Consideraciones

- Lenguaje libre (queda a su discreción)
- Para las estructuras no se pueden usar librerías o APIs
- Usar únicamente librerías estándar y no librerías que funciones en un único sistema operativo.
- Entrega 29 de Abril del 2024
- Subir enlace de git a Classroom
 - Si no se puede descargar, se considerará como no entregado.

Requerimientos

- Manual Técnico
- Manual de Usuario
- Interfaz Gráfica amigable