

Algoritmos y Programación II (75.41)

Trabajo Práctico N°3

26/11/2012

1. Consigna

La famosa pizzería de Gerli continúa su proceso de expansión. El último paso es mejorar su servicio de delivery, ya que muchas veces el cliente iba a pagar el pedido al local pero la comida nunca llegaba. Para lograr esto, nos han pedido que desarrollemos un sistema que encuentre la mejor ruta entre la pizzería y el domicilio del cliente, y genere un mapa para que el repartidor pueda llegar a hacer la entrega, y volver una vez entregadas las pizzas.

El repartidor debe llegar lo más pronto posible al destino, por lo que el tiempo que tarda es más importante que la distancia recorrida. Dado que, por ej., circular por una avenida es más rápido que por una calle, el recorrido debería desviarse para pasar por la avenida (obviamente, siempre que el tiempo que toma desviarse y circular por la avenida sea menor al tiempo que tardaría al ir por una calle).

Para poder buscar la dirección y calcular el camino se nos provee un archivo que tiene listadas todas las intersecciones de calles del partido de Avellaneda, de donde es oriunda la pizzería, y toda la información de dichas calles (tipo de calle, sentido de circulación, etc.). El archivo tiene el siguiente formato:

```
N
nodo 1,x,y,longitud,latitud
nodo 2,x,y,longitud,latitud
...
nodo N,x,y,longitud,latitud
M
calle 1,nombre,tipo de calle,mano única,nodo inicial,nodo final
```

```
calle 2,nombre,tipo de calle,mano única,nodo inicial,nodo final
...
calle M,nombre,tipo de calle,mano única,nodo inicial,nodo final
```

Donde:

- N: Cantidad de intersecciones.
- x e y: Coordenadas locales, usadas para calcular la distancia entre una intersección y otra.
- latitud y longitud: Coordenadas reales, para armar la ruta.
- M: Cantidad de tramos de calle.
- tipo de calle: número 1-3, siendo 1 avenida principal, 2 avenida secundaria, y 3 calle.
- mano única: será un 0 si es doble mano, y 1 si es de sentido único. En caso de ser de sentido único, se recorre desde el nodo inicial al final.
- nodo inicial: número de la intersección en la que comienza el tramo de calle.
- nodo final: número de la intersección en la que finaliza el tramo de calle.

Es importante notar que las intersecciones no siempre son entre dos calles solamente: hay intersecciones con mayor cantidad de esquinas, hay calles sin salida, calles que debido a un cambio de dirección se intersecan consigo mismas, etc. Además, notar que hay intersecciones que se dan más de una vez; en casos donde haya más de una intersección posible para dos calles, elegir cualquiera de ellas arbitrariamente.

El programa recibirá la ruta del archivo, y cargará toda la información en memoria de la manera que sea más conveniente. Al principio, y por única vez, pedirá la dirección de la pizzería. Hasta que se decida terminar la ejecución del programa, se pedirá ingresar la intersección de calles que corresponde a la dirección del cliente. En caso de que no exista o esté mal ingresada, debe informarse por pantalla y volver a pedir la dirección de envío.

Se calculará la ruta óptima entre la pizzería y la intersección dada, tanto de ida como de vuelta, informando por pantalla la distancia recorrida en cada caso, y cargando dichas rutas en un archivo KML. Una vez procesado el pedido, se puede ingresar la dirección de un nuevo pedido.

Una vez finalizada la ejecución, se debe cerrar el archivo KML que contiene todas las rutas calculadas.

El formato y la información necesaria para generar dicho archivo se encuentran en el archivo de ejemplo que está en la página de la materia.

1.1. Utilización de estructuras de datos

Las intersecciones y los tramos de calle que las unen deben modelarse con un TAD Grafo, según lo visto en clase. Es importante notar, según lo aclarado sobre las intersecciones de calles, que todos los nodos tendrán al menos una arista que converja a ellos. Con respecto a las aristas, tener en cuenta que varias de ellas pueden tener el mismo nombre.

Además del grafo, deben usar los TAD vistos hasta el momento o hacer nuevos, según sea necesario.

El código para este sistema puede desarrollarse en cualquier lenguaje que sea multiplataforma (previo acuerdo con el ayudante asignado al grupo).

2. Consignas opcionales

- A veces (casi siempre) el cliente queda insatisfecho con el servicio. Por temor a algún tipo de represalia o venganza por parte de los mismos, los repartidores evitan las intersecciones en las que hicieron entregas. Se pide que al calcular una ruta se excluyan las intersecciones a las que se haya hecho una entrega previamente, de manera que la misma no pase por éstas.
- Como ya se sabía, las motos pueden llevar hasta 5 pizzas. Para aprovechar los viajes, se pide que se de la opción de calcular la ruta a seguir para entregar pizzas para hasta 5 clientes diferentes en el mismo viaje.
- Al realizar varias entregas en un solo viaje puede que la ruta no sea la mejor, y tardar más innecesariamente. Se pide que al calcular la ruta para realizar varias entregas en un viaje (ítem anterior), dicha ruta sea la óptima.
Sugerencia: Usar backtracking para encontrar la mejor entre todas las posibles rutas.
- Las cosas siempre pueden fallar, y los pedidos confundirse (aunque solo haya una variedad de pizza). Nos comentaron que sería bueno añadir la funcionalidad de poder calcular la mejor ruta entre dos puntos, de manera que cuando el cliente llame para quejarse se le pueda explicar como ir a la casa del otro cliente para intercambiar los pedidos. Se pide poder calcular la ruta óptima entre dos puntos cualesquiera, y no solo entre la pizzería y un punto.

- Los repartidores usan sus motos para otras cosas, no siempre del todo legales, además de para repartir pizza. Por esa razón las mismas no se encuentran en el mejor estado, y no soportan viajes muy largos. Permitir ingresar la distancia máxima que puede tener la ruta antes de calcular la misma e informar por pantalla si no es posible encontrar un camino con esa limitación.

Para la aprobación del Trabajo Práctico se exige al menos una de las consignas opcionales. Además, todo el trabajo adicional será tenido en cuenta en la calificación final.

3. Criterios de aprobación

A continuación describimos criterios y lineamientos que deben respetarse en el desarrollo del trabajo

3.1. Programa

El programa debe cumplir los siguientes requerimientos:

- Debe estar adecuadamente estructurado y modularizado, utilizando funciones definidas de la forma más genérica posible, sin caer en lo trivial.
- El código debe ser claro y legible.
- El código debe estar comentado y las funciones definidas, adecuadamente documentadas.
- De utilizarse un lenguaje compilado, el programa debe compilar sin mensajes de error (`-Wall -pedantic`), debe correr sin pérdidas de memoria, uso de valores sin inicializar, o errores en general. Es decir que, el programa debe correr en valgrind sin errores.
- Además, claro, debe satisfacer la especificación de la consigna.

3.2. Informe

El informe deberá consistir de las siguientes partes:

- Carátula con la información de los integrantes del grupo y el ayudante asignado.
- **Análisis, especificación y diseño:** describir el problema, cuál es la solución y cómo se lleva a cabo. Incluir las convenciones de código utilizadas (criterios para nombres de variables y funciones, criterios de indentación, etc).
- **Implementación:** Incluir aquí *todo* el código fuente utilizado en formato **monoespaciado**, para facilitar su lectura.
- **Pruebas unitarias:** probar las funcionalidades de cada estructura o porción del programa.
- Mantenimiento (*Opcional*): cambios importantes realizados a lo largo del trabajo.

- También *opcionalmente*, toda explicación adicional que consideren necesaria, referencias utilizadas, dificultades encontradas y conclusiones.

El informe debe estar lo más completo posible, con presentación y formato adecuados. Por ejemplo, este enunciado cumple con los requerimientos de un informe bien presentado.

4. Entrega

El trabajo consiste en:

- El informe impreso.
- El informe digital, en formato `.pdf`
- Una versión digital de **todos** los archivos de código fuente, separados del informe, en un archivo comprimido (`.zip` o `.tar.gz`).

Los dos últimos deben enviarse a la dirección `tps.7541rw@gmail.com`, colocando como asunto: **TP3 - Padrón1 - Apellido1 - Padrón2 - Apellido2**.

El informe impreso debe entregarse en clase. El plazo de entrega vence el 7/12/2012.