

Examen 1

1. Descripción del problema

Este problema está basado en un problema de la ICPC [1]. Se quiere que desarrolle una aplicación para la agencia de turismo *USB Tours*. Se tiene que *USB Tours* ofrece a turistas un servicio que consiste en un tour guiado por varias ciudades del país. Un tour consiste en un recorrido desde una ciudad de partida hasta una ciudad de llegada. Cada tour es realizado por un grupo de turistas, los cuales están acompañados por un guía de *USB Tours*. Para viajar desde una ciudad x hasta una ciudad y se usa algún servicio de transporte de la ciudad x , el cual tiene una limitación en el número de personas que puede llevar en el transporte disponible. En consecuencia, es posible que no todos los turistas puedan viajar desde una ciudad a otra. En caso de que no todos los turistas puedan viajar desde la ciudad de partida hasta la ciudad de llegada, en un solo tour, el guía hace lo siguiente. El guía encuentra en un mapa, la ruta desde la ciudad de partida hasta la ciudad de llegada, que permita la movilización de turistas, en el menor número de tours posible. Si no todos los turistas pueden ser transportados en un tour, entonces el turista primero hace un tour con un grupo de turistas. Luego se regresa hasta la ciudad de partida y hace otro tour con otro grupo de turistas. Y así va haciendo sucesivamente tours hasta que haya transportado a todos los turistas hasta la ciudad de llegada. Los caminos entre ciudades son bidireccionales. El objetivo es que usted realice una aplicación que dado un mapa, una ciudad de partida, una ciudad de llegada, y el número de turistas, retorne una ruta desde la ciudad de partida, hasta la ciudad de llegada, tal que a través de la misma, sea posible movilizar los turistas haciendo el menor número de tours posible.

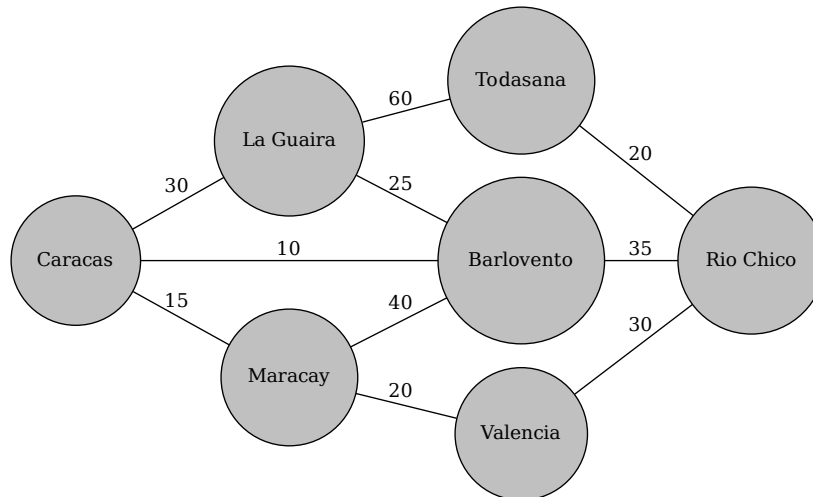


Figura 1: Mapa de ciudades. El valor que se encuentra en la conexión entre las ciudades corresponde al número máximo de personas que pueden viajar en un medio de transporte, desde una ciudad hasta otra.

Por ejemplo, dado el mapa de la Figura 1, se tiene que se *USB Tours* quiere llevar 99 turistas

desde *Caracas* hasta *Ri  Chico*. La ruta que permite transportar los turistas en el menor n mero de viajes es *Caracas – La Guaira – Barlovento – Ri  Chico*. El n mero de tours realizados en esa ruta es 5. Tome en cuenta que el gu a ocupa un asiento en los transportes, al igual que los turistas, en cada uno de los tours.

La idea es que para implementar la soluci n del problema, debe modelar el mismo como un grafo y utilizar los algoritmos sobre grafo vistos en clase, para su soluci n. El programa cliente que resuelve el problema debe hacer uso de la librer a *libGrafoKt*. La librer a puede modificarse y/o ampliarse para poder resolver el problema planteado. No se aceptar  soluciones adaptadas a este problema que no usen la librer a *libGrafoKt*.

1.1. Entrada de los datos

El programa debe realizarse en Kotlin, y debe estar contenido en el archivo llamado `USBTours.kt`. La ejecuci n del programa debe hacerse por medio de un *shell script* llamado `runUSBTours.sh`. Adem s debe crear un archivo Makefile que sea capaz de compilar el c digo entregado. El archivo `runUSBTours.sh` se ejecuta con la siguiente l nea de comando:

```
>./runUSBTours.sh <inicio> <llegada> <turistas> <mapa>
```

Donde:

inicio: La ciudad de inicio.

llegada: La ciudad de destino.

turistas: Un n mero entero que corresponde al n mero de turistas a transportar.

mapa: Archivo con el mapa. El formato del archivo es como sigue. La primera l nea indica el n mero N de ciudades, la segunda l nea indica el n mero M de conexiones entre ciudades. Luego siguen M l neas, en la que se tiene el nombre de dos ciudades conectadas por un camino y el n mero de personas que pueden viajar de una ciudad a otra. Estos tres elementos est n separados por una coma.

1.2. Salida de los datos

Se debe imprimir por la salida est ndar una primera l nea con la ruta que permite hacer el menor n mero de tours, como secuencia ciudades separadas por un gui n. Luego se debe imprimir una segunda l nea con el con el m nimo n mero de tours que se deben realizar para transportar a los turistas desde la ciudad de partida, hasta la ciudad de llegada.

1.3. Ejemplo pr ctico

Suponga que se tiene un mapa contenido en el archivo `miMapa.txt`, con el siguiente contenido:

```
7
10
Caracas,La Guaira,30
Caracas,Barlovento,10
Caracas,Maracay,15
La Guaira,Barlovento,25
Maracay,Barlovento,40
```

```
La Guaira,Todasana,60
Todasana,Rio Chico,20
Barlovento,Rio Chico,35
Maracay,Valencia,20
Valencia,Rio Chico,30
```

Se ejecuta la aplicación con la siguiente línea de comando:

```
>./runUSBTours.sh Caracas "Rio Chico" 99 miMapa.txt
```

El programa debe mostrar por la salida estándar lo siguiente:

```
Caracas-La Guaira-Barlovento-Rio Chico
5
```

2. Condiciones de entrega

El código del examen debe estar contenido en una carpeta llamada *examen1*. Esta carpeta debe contener todos los archivos fuentes del examen, y un archivo Makefile para compilar el código del mismo. La carpeta *examen1* y la declaración de autenticidad debidamente firmada, deben estar contenidos en un archivo comprimido, con formato *tar.xz*, llamado *Examen1_X.tar.xz*, donde *X* es el número de carné de los estudiante. La entrega del archivo *Examen1_X.tar.xz*, debe hacerse por medio de la plataforma *Classroom* antes de las 12:30 P.M. del 01 de julio de 2022.

Referencias

- [1] International Collegiate Programming Contest, “Past problems,” 2022, [Online; accessed 24-Junio-2021]. [Online]. Available: <https://icpc.global/worldfinals/problems>