



## Trabajo Práctico - Mesas de Julio + Flotante de agosto

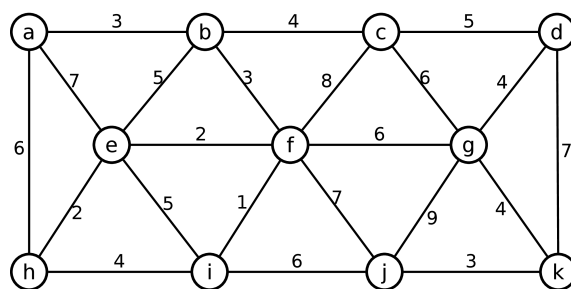
### 1. Motivación del problema

El objetivo de este trabajo práctico consiste en proponer una estructura de datos y un algoritmo que permita encontrar una solución, siempre que sea posible, para el **TSP** ("Traveling Salesman Problem").

El problema es tal que dado un conjunto finito de ciudades, y costo de movilidad entre pares de ellas (lados), se desea encontrar la forma más económica de visitar todas las ciudades exactamente una vez, y volver al punto de partida.

Los primeros estudios del problema del vendedor viajante se sitúan alrededor del 1830 pero, todo parece indicar que la forma general del **TSP** fue estudiada, por primera vez, por matemáticos en Viena y Harvard, durante los años 1930s.

Veamos un ejemplo de esto. Considere el siguiente diagrama:



en el mismo se puede ver que las ciudades son  $\{a, b, c, d, e, f, g, h, i, j, k\}$  y los costos están dados por un valor entero, estrictamente mayor a 0, dado. Por ejemplo, el costo de ir de  $a$  a  $b$  (ó de  $b$  a  $a$ ) es 3.

Asumiremos que entre cada par de ciudades puede haber, a lo sumo, un lado que las una. En el ejemplo anterior, las ciudades  $a$  y  $f$  no están unidas directamente pero puedo llegar de  $a$  hasta  $f$ , por ejemplo, pasando por  $b$ .

Es importante remarcar que los datos de entrada que recibiremos siempre contarán con un camino como el pedido, es decir, nuestro problema siempre tendrá solución.

### 2. Objetivos

El programa que se tendrá que realizar, deberá tomar un archivo de entrada con la información de las ciudades y el costo de los caminos y deberá generar una salida con la ruta, es decir, la secuencia de ciudades a realizar y el costo, es decir, la suma de los lados que la forman.

### 3. Detalles de la implementación

El método main debería permitir pasar el nombre del archivo de entrada, el nombre del archivo de salida. Es decir, sería algo así:

main archivoEntrada archivoSalida

### 3.1. Formato de archivo de entrada y salida

El archivo de entrada tendrá una lista de ciudades, tomaremos como identificador de las ciudades siempre strings sin espacios en blanco, separadas por coma y, una lista de lados con sus costos. La salida debe ser una salida de los lados con los costos que forman la ruta de menor costo.

### 3.2. Ejemplo de entrada y salida

Entrada:

Ciudades

a, b, c, d, e, f, g, h, i, j, k

Costos

a,b,3

b,c,4

a,e,7

a,h,6

...

Salida:

a,b,3

b,f,1

...

## 4. Evaluación

Para la evaluación del Trabajo Práctico se tomarán en cuenta los siguientes elementos:

- a) Estructura de datos usada
- b) Algoritmo propuesto
- c) Eficiencia
- d) Calidad del código entregado