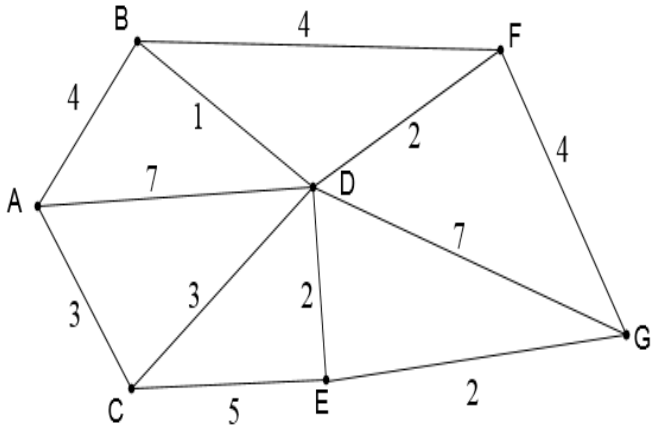
Proyecto # 2: La ruta más eficiente.

Imaginemos a un técnico de instalación de cableado eléctrico que tiene que conectar una red eléctrica entre varios pueblos para enviar cargas eléctricas de un pueblo a otro de la red. Su jefe, motivado a la situación actual del país, le obliga a utilizar la menor longitud de cable posible para ahorrar gastos. Adicionalmente, es necesario determinar la Ruta Más Eficiente para enviar una carga de un pueblo a otro. Los pueblos están conectados de la manera siguiente:



- Cada arista del grafo representa el camino más directo entre los dos pueblos de los extremos.
- Cada vértice (nodo) representa las plantas eléctricas de cada uno de los pueblos.
- Los números representan la distancia entre cada uno de ellos

Para conectar todos los puntos usando la menor longitud total de cable posible, el técnico utilizará la combinación de distintos algoritmos (Floyd, Prim o Kruskal), es decir, para encontrar el **árbol recubridor mínimo (ARM)** en grafos conexos dirigidos (o no) cuyas aristas tienen peso. Mediante esto, el técnico de instalación eléctrica podría rentabilizar el material empleado en el proceso y así rebajar el coste de producción.

Ahora, para poder enviar una carga desde un punto de la red a otro específico de forma **eficiente** también debería ser utilizado un algoritmo específico (por ejemplo Dijsktra), sobre el árbol recubridor mínimo (ARM) para obtener la ruta más corta (RMC), la cual vendría a ser la Ruta Más Eficiente.

Se desea que Ud. ayude al técnico a determinar la ruta más eficiente mediante la programación de los métodos que satisfagan los requerimientos. Para ello será necesario utilizar el lenguaje de programación ***Haskell***.

Entrada y Salida.

Para esto se le facilitará un archivo de texto (ADY.TXT) con la matriz de adyacencia ponderada que define el grafo y los nombres de los pueblos a ser conectados mediante una emisión de carga.

(A B C D E F G)
[0 4 3 7 0 0 0]
[4 0 0 1 0 4 0]
[3 0 0 3 5 0 0]
[7 1 3 0 2 2 7]
[0 0 5 2 0 0 2]
[0 4 0 2 0 0 4]
[0 0 0 7 2 4 0]
[A G]

En caso que no puede implementar la entrada por medio de archivos, debe especificar la entrada de datos manual.

La salida se emitirá por salida estándar y consistirá en dos listas: la primera representando el **árbol recubridor mínimo** del grafo y la segunda, la **ruta más eficiente** solicitada para enviar la carga.

Entrega.

La entrega se realizará mediante la plataforma Google Classroom. Se consignará el código fuente en Haskell (.hs), **autodocumentado**, sin comprimir.