

IMPORTANTE: Las tareas son **individuales**. Cualquier acción que pueda beneficiar de forma injusta la calificación de su tarea está prohibida, incluyendo la presentación de cualquier componente que no es de su autoría o la facilitación de esto para otros. Es considerado aceptable discutir -en líneas generales- los métodos y resultados con sus compañeros, pero se prohíbe explícitamente realizar las tareas en conjunto o compartir código de programación.

Grafos

Un **router**, también conocido como enrutador o encaminador, es un dispositivo de red que renvía paquetes de datos entre dos sub-redes. Un **router** se conecta a dos o más líneas de datos. Cuando un paquete de datos es recibido, el **router** lee la información de dirección para determinar su destino final. Utilizando información en su **routing table**, redirecciona el paquete a la red siguiente. Este es el principio de funcionamiento de **internet**.

Para asegurar un servicio constante y garantizado de transmisión de datos entre dos puntos, la red a construir debe ser altamente redundante. Esto es, existen múltiples rutas para poder enviar un paquete de datos desde un punto a otro, y la estrategia finalmente a utilizar dependerá, entre otras cosas, del tráfico que incide en la velocidad de envío asignado a cada línea de transmisión (que son bi-direccionales).

Trabajo a realizar

1. Realice un programa que simule una conexión entre 1000 **routers** que están interconectados. La matrix de conexión entre los routers viene data en el archivo `connectionMatrix.txt`, donde la conexión es **bidireccional**. Cada **router** posee una dirección dada por un identificador de 4 dígitos¹. Cada línea de enlace entre dos routers tiene asignada una velocidad de transmisión dada en Gbits/s. La velocidad de transmisión de cada línea se obtiene de manera aleatoria al inicio de su programa entre 0 y 100Gbits/s.

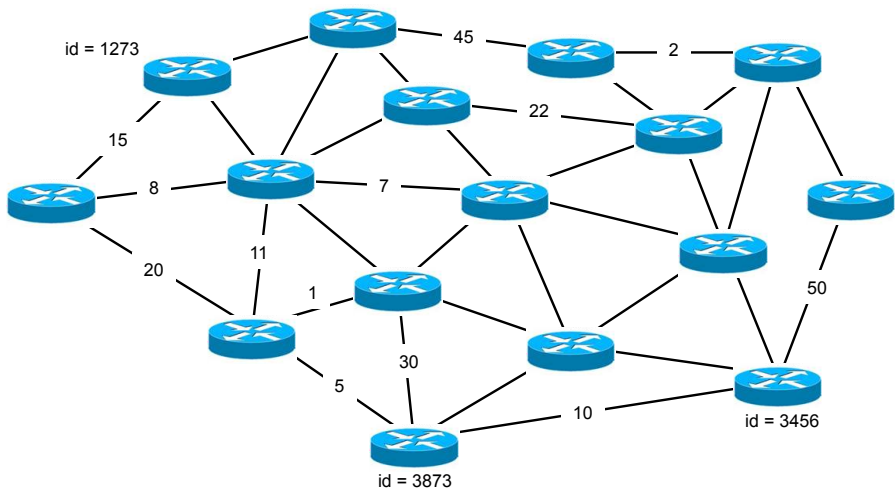


Figura 1: Red de ejemplo para representar las conexiones en una red de datos. Los ids de los routers son números de 4 dígitos. Las velocidades de transmisión de las líneas de conexión entre routers está especificada en Gbits/s.

2. Su código debe generar como salida un archivo de texto (`network-config.txt`) de 1000 filas por 1000 columnas con la matriz de adyacencia de la red formada por los routers, pero en este caso usando los valores de las velocidades de transmisión. (Donde había un 1 anteriormente, ahora se verá reemplazado por el valor de la velocidad de transmisión del enlace correspondiente).

¹En la realidad cada **router** tiene una dirección IP: [0..255].[0..255].[0..255].[0..255]

3. Con este grafo genere el árbol de cobertura mínimo (i.e., mininum-spanning tree, subgrafo mínimo) para definir así la ruta por defecto (que no siempre es la más corta) que un paquete de datos debe seguir. Para esto ocupe el algoritmo de Kruskal.
4. Además, imprima por pantalla la lista de conexiones del árbol de cobertura mínima ordenadas por costo (incluyendo los dos vértices y el costo).

Condiciones de Entrega

Utilice `/*comentarios*/` para describir lo que se hace en cada etapa de su código. Prepare un archivo adicional README.txt donde debe especificar sus datos (nombres, apellidos, ROL USM, y emails) y ademas que es cada archivo de su programa, como debe ser compilado y como debe ser ejecutado. Ademas, debe explicar cualquier particularidad que pueda tener su programa. El código deberá estar escrito según el estándar de codificación GNU y ser compilado y ejecutado en el servidor Aragorn **SIN ERRORES** para ser evaluado. El código escrito debe estar indentado de forma coherente y consistente. Para la entrega, envíe un solo archivo (.zip, .rar, .tar.gz, etc.) que contenga su código, es decir, sus archivos .c, .h utilizados, mas el archivo README.txt. El entregable debe ser subido a la plataforma AULA antes de las 23:50 del viernes 25 de marzo 2020. Se aplicara descuento de Fibonacci por hora (i.e., 1 punto la primera hora, 1 punto la segunda, 2, 3, 5, 8, 13, 21, 33, 54, etc..).