



Esta parte del proyecto consiste en hacer una implementación de las funciones que se describen a continuación. Se recomienda utilizar **MATLAB** u **Octave** (un programa similar a **MATLAB** pero con licencia GPL), pero se puede usar otro lenguaje. En caso de usar **MATLAB**, el código **debe ser compatible** con **Octave**. En caso de usar otro lenguaje, se debe proveer al profesor de todas las herramientas necesarias para compilar y ejecutar el programa, así como de instrucciones para hacerlo.

1. Escribir una función que lea los datos de una red y los almacene en una estructura combinada estrella directa-reversa. Los datos vienen en un archivo de texto en un formato de entrada basado en el de DIMACS. Este formato está organizado por líneas. Cada línea comienza con una letra que identifica el propósito de dicha línea y los datos están separados por al menos un espacio en blanco. A continuación se describen los tipos de línea.
 - a) **Líneas de comentario.** Como su nombre lo indica, estas líneas sirven para añadir comentarios al archivo y deben ser ignoradas por los programas. Pueden aparecer en cualquier lugar del archivo. Cada una comienza por la letra “c” (en minúscula) y a continuación puede contener cualquier cosa.
 - b) **Línea de problema.** Hay una sola de éstas por archivo y debe aparecer antes de cualquier línea de nodo o arco. Comienza por la letra “p” (en minúscula). El segundo dato tiene tres letras y debe ser ignorado. Los siguientes dos datos son dos enteros: el número de nodos y el número de arcos de la red.
 - c) **Líneas de nodos.** Estas líneas comienzan por la letra “n” (en minúscula). El siguiente dato es un entero que indica el número del nodo y el último un entero que indica la oferta de dicho nodo. De no ser necesarias las ofertas o si éstas son cero, las líneas correspondientes pueden ser omitidas. De aparecer, deben hacerlo antes de cualquier línea de arco.
 - d) **Líneas de arcos.** Comienzan por la letra “a” (en minúscula). Los siguientes dos datos representan la cola y la cabeza del arco. A continuación aparecen el flujo, la capacidad y el costo del arco, en ese orden.
2. Escribir una función que use la estructura estrella directa-reversa leída por la función anterior y encuentre la lista de arcos que salen de un nodo i ($A^+(i) = A(i)$) y la lista de arcos que llegan a ese nodo ($A^-(i)$). Esta función tendrá como parámetro de entrada el número del nodo.
3. Escribir una función que use la misma estructura estrella directa-reversa y encuentre una descomposición del flujo en caminos y ciclos dirigidos según el algoritmo cubierto en clase.

El código entregado deberá incluir una interfaz sencilla para poder ser ejecutado por el usuario. La salida debe ser claramente legible mostrando cada uno de los resultados requeridos de las funciones.

Este trabajo se **debe hacer** en equipos de dos personas, especificando claramente el rol de cada uno de los integrantes. Cada equipo debe trabajar independientemente de los otros, es decir, no se permite el intercambio de información entre los equipos. Se debe enviar el código por e-mail a bfeijoo@usb.ve¹ o entregarlo en un diskette o CD. Dicho código debe estar documentado, **incluyendo instrucciones para su ejecución**. Es responsabilidad de cada equipo asegurarse de que sus programas funcionen correctamente. El profesor realizará pruebas con archivos *proprios*.

Esta parte del proyecto vale 10 puntos y debe ser entregada, a más tardar, el lunes 8 de noviembre de 2010.

¹El servidor de la USB rebota los archivos con extensión “zip”. En caso de que necesite enviar un archivo de este tipo, basta cambiar la extensión, enviarlo y avisar al profesor acerca del cambio.