



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 el formato que se usó en la parte I, con las modificaciones que se citan a continuación:
 - a) Cada línea de nodo, aparte del número del nodo y la oferta, contiene otros tres números enteros que representan las componentes, para ese nodo, de los arreglos **pred**, **depth** y **thread**, en ese orden.
 - b) Cada línea de arco contiene, aparte de los datos mencionados en la parte I, un entero adicional que es -1 si el arco está en \mathcal{L} , 1 si está en \mathcal{U} y 0 si está en \mathcal{T} .

Se puede suponer que la estructura dada corresponde a una SAE factible.

2. Escribir una función que use la estructura estrella directa-reversa leída por la función anterior y realice las siguientes tareas:
 - a) Obtener y mostrar los potenciales π y los flujos x correspondientes a esta estructura.
 - b) Calcular y mostrar c^π para los arcos en \mathcal{L} y en \mathcal{U} .
 - c) Sugerir al usuario los posibles arcos entrantes (k, ℓ) y permitirle elegir uno.
 - d) Obtener y mostrar el arco saliente (p, q) .
 - e) Actualizar y mostrar los nuevos π y x .

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 15 puntos y debe ser entregada, a más tardar, el lunes 6 de diciembre 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.