



# Traductor de grafos dirigidos

## Laboratorio Semana 3

### 1 Introducción

La recién formada Grafos y Algoritmos SA, es el resultado de la unión entre Grafos SRL y Algoritmos CA. Ambas compañías trabajan con grafos dirigidos; sin embargo, Grafos SRL almacenaba sus archivos de grafos en el formato “Lista de Adyacencias”, mientras Algoritmos CA almacenaba sus grafos en el formato “Matriz de Adyacencias”. Estos formatos significan que los siguientes ambos representan el mismo grafo:

0: 3	0 1 2 3 4 5 6
1: 2	-----
2: 3	0  0 0 0 1 0 0 0
3: 5	1  0 0 1 0 0 0 0
4: 1 2 3	2  0 0 0 1 0 0 0
5: 0 2 6	3  0 0 0 0 0 1 0
6: 0 1 4	4  0 1 1 1 0 0 0
	5  1 0 1 0 0 0 1
	6  1 1 0 0 1 0 0
Lista de adyacencias	Matriz de adyacencias

### 2 Descripción de la Actividad

Se le ha contactado para que desarrolle un programa capaz de leer grafos en un formato y traducirlos al otro. El programa debe cargar el archivo en una representación interna y luego realizar la conversión. Debido a que Algoritmos CA es el accionista mayoritario, se ha acordado que la representación interna de este programa al momento de guardar debe ser una Matriz de Adyacencias similar a la mostrada en la sección 17.3 de libro [Algorithms in Java](#) de Robert Sedgewick. La representación traducida se debe imprimir por consola.

Puede usar las clases [FileReader](#), [BufferedReader](#), y [String](#) de Java para procesar el archivo. Observe que, con el formato de la primera línea, se puede deducir el tipo de archivo. Observe también que, en el caso de la matriz de adyacencias, el número de vértices viene dado por la cantidad de números separados en esta primera fila, mientras que, en el caso de la lista de adyacencias, no es conocido hasta llegar a la última fila; por lo que se debe cargar en una representación intermedia que crezca dinámicamente, tales como las subclases de [Collection](#). También puede ser útil consultar [System.arraycopy\(...\)](#)

### 3 Entrada de Datos

El programa se debe poder ejecutar desde la consola con el siguiente comando:

```
java Cliente <archivo>
```

donde <archivo> es la ruta del archivo de grafo que se desea traducir

#### 3.1 Formato “Lista de Adyacencias”

Cada línea de un archivo “lista de adyacencias” se compone de un identificador de vértice numérico  $v_i$  entre 0 y  $n - 1$ , seguido de dos puntos, y una lista de  $m$  vértices a los cuales existen arcos desde  $v_i$  de la siguiente forma:

$$v_i : \cdot v_1 \cdot v_2 \cdot \dots \cdot v_m$$

Los interpuntos ( $\cdot$ ) representan los espacios

Siempre hay exactamente  $n$  líneas.

#### 3.2 Formato “Matriz de Adyacencias”

Los archivos “matriz de adyacencias” comienzan siempre con tres espacios en blancos, seguidos de un identificador numérico entre 0 y  $n - 1$  para cada uno de los  $n$  vértices separados por espacios. En lo siguiente, los interpuntos ( $\cdot$ ) representa los espacios

$$\cdot \cdot \cdot v_1 \cdot v_2 \cdot \dots \cdot v_n$$

Puede usar “\w” para atrapar todo tipo de espacios; así, su programa funcionará aunque los vértices estén separados por tabuladores.

La segunda línea es siempre una división horizontal formada por  $2n+2$  guiones (-)

Luego, hay  $n$  líneas de la forma

$$v_i \cdot | \cdot v_i v_1 \cdot v_i v_2 \dots v_i v_n$$

donde  $v_i v_j$  es el número de arcos que van del  $i$ -ésimo vértice al  $j$ -ésimo vértice

Si el formato no se cumple (por ejemplo, hay menos de  $n$  líneas) debe arrojar un error.

### 4 Entrega

Debe entregar como mínimo

- TraductorGrafo.java, una clase abstracta que contenga los métodos y variables comunes de sus clases hijas
  - TraductorDesdeLista.java, una clase concreta cuyos métodos permitan traducir archivos en el formato Lista de Adyacencias al formato Matriz de Adyacencias
  - TraductorDesdeMatriz.java, una clase concreta cuyos métodos permitan traducir archivos en el formato Matriz de Adyacencias al formato Lista de Adyacencias
- Cliente.java, el programa que llama a los métodos necesarios para realizar la traducción

A través del Moodle antes del lunes, 3 de febrero a las 11:55 pm (lunes de semana 5). Todos los métodos y variables que agreguen deben estar debidamente documentados.

## 5 Evaluación

De ser escogido para evaluación, se asignarán:

- 5 puntos por código
  - Métodos: 1 punto por cada una de las clases mencionada anteriormente
  - Estructuras de datos: 1 punto por su representación del grafo
- 4 puntos por ejecución
  - 2 puntos por poder convertir de Matriz de Adyacencias a Lista de Adyacencias
    - 1 punto por poder hacerlo cuando los vértices estén en orden
    - 1 punto por poder hacerlo cuando los vértices no estén ordenados
  - 2 puntos por poder convertir de Lista de Adyacencias a Matriz de Adyacencias
    - 1 punto por poder hacerlo cuando los vértices estén en orden
    - 1 punto por poder hacerlo cuando los vértices no estén ordenados
- 1 punto por documentación

Se asignará un punto adicional si logran detectar si el grafo representado es un grafo no-dirigido. Un grafo no-dirigido representa a un grafo dirigido, si para cada arco  $v_i v_j$  existe el correspondiente arco  $v_j v_i$ . Esta detección debe imprimirse luego de imprimir el grafo traducido y no debe ser parte de la función `imprimirGrafoTraducido`.