

## TAREA N°2 PROGRAMACIÓN EN C: GRAFOS.

### I. OBJETIVOS.

El presente enunciado tiene como objetivo: Diseñar e implementar operaciones sobre grafos a través de su representación matricial.

### II. INTERFAZ.

El programa deberá solicitar al usuario el nombre de un archivo de texto que debe contener la información del grafo que se desea procesar. El formato del archivo es:

```
númeroNodos númeroArcos tipoGrafo
nodo nodo costo
nodo nodo costo
nodo nodo costo
...
```

Donde el tipoGrafo corresponderá a un **1** si se trata de un **grafo dirigido** y a un **2** si se trata de un **grafo no dirigido**. Las siguientes líneas contendrán la información de cada arco del grafo, indicando para aquello nodo de inicio del arco, no de llegada del arco, y costo del arco. En la figura 1 puedes observar un ejemplo de este formato.

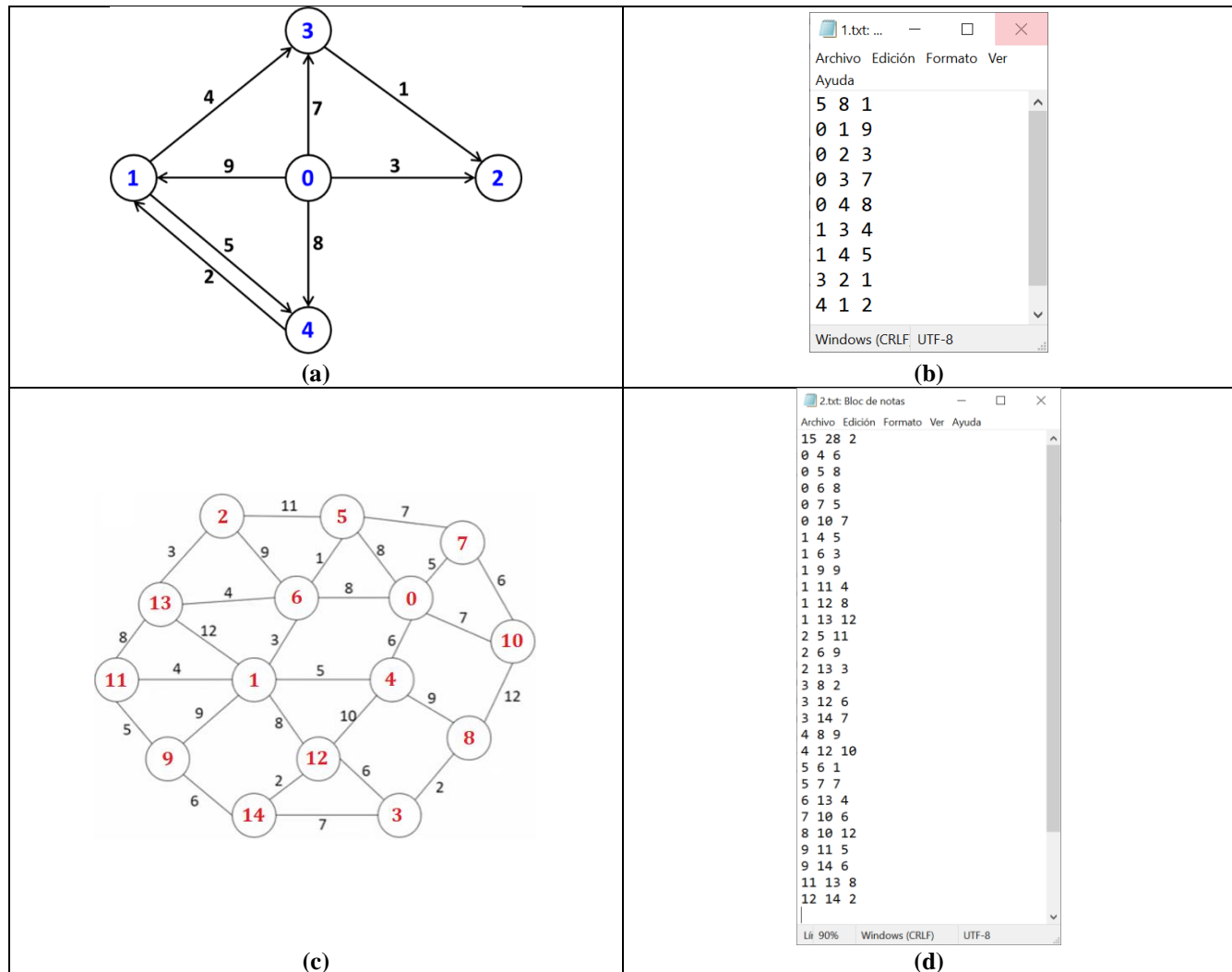
Las funcionalidades que debes implementar sobre el grafo recibido dependerán del tipo de grafo que corresponde (algunas coinciden en ambos tipos):

#### II.1 Funcionalidades solicitadas para un grafo no dirigido:

- Imprimir la matriz de costos del grafo.
- Imprimir la matriz de su grafo complemento.
- Imprimir el arco de mayor costo y su costo.
- Imprimir los adyacentes y grado para cada nodo del grafo.
- Imprimir si se trata de un grafo conexo o desconexo. Si es desconexo debe indicar además el número de componentes conexas.
- Imprimir si se trata de un grafo completo o no.
- Imprimir si se trata de un grafo k-regular o no. Si es k-regular debes imprimir el valor de k.
- Imprimir el recorrido en profundidad desde cada nodo.
- Imprimir el recorrido en amplitud desde cada nodo.
- Imprimir los caminos de costo mínimo desde cada nodo (usando Dijkstra) a todos los otros. Si no existe un camino el programa debe imprimir un mensaje.

#### II.2 Funcionalidades solicitadas para un grafo dirigido:

- Imprimir la matriz de costos del grafo.
- Imprimir el arco de mayor costo y su costo.
- Imprimir los sucesores y grado de salida para cada nodo del grafo.
- Imprimir los antecesores y grado de entrada para cada nodo del grafo.
- Imprimir el recorrido en profundidad desde cada nodo.
- Imprimir el recorrido en amplitud desde cada nodo.
- Imprimir los caminos de costo mínimo desde cada nodo (usando Dijkstra) a todos los otros. Si no existe un camino el programa debe imprimir un mensaje.



**Figura 1:** Ejemplos del formato del archivo de entrada con el dibujo del grafo asociado. En (a) aparece un grafo dirigido, y en (b) se muestra cómo se debe almacenar en un archivo de texto. En (c) podrás observar un grafo no dirigido y en (d) su correcto almacenamiento en un archivo de texto.

### III. LIBRERÍA DE APOYO Y CÓDIGO FUENTE DE PARTIDA.

Para apoyarte en el desarrollo de esta tarea se te entregarán 2 archivos: Un código fuente para que inicies tu tarea (llamado “**código base.c**”) que ya posee la solicitud del nombre del archivo y su lectura y muestra por pantalla la matriz del grafo, y una librería llamada “**funciones.h**” que posee las funciones que se describen a continuación:

- **creaArregloBool**: retorna un arreglo de n de booleanos, inicializado en falsos.
- **creaArreglo**: retorna un arreglo de n de enteros, inicializado en 0s.
- **creaMatriz**: retorna una matriz cuadrada de nxn de enteros, inicializada en 0s.
- **imprimeMatrizGrafo**: imprime por pantalla la matriz “Grafo” de nxn enteros, que recibe como parámetros.
- **Profundidad**: Imprime por pantalla el recorrido en profundidad desde un nodo a través de un grafo, ambos recibidos como parámetros.
- **Amplitud**: Imprime por pantalla el recorrido en profundidad desde un nodo a través de un grafo, ambos recibidos como parámetros.

- Dijkstra: Imprime por pantalla todos los caminos mínimos desde un nodo  $v_0$  a través de un grafo recibidos como parámetros.

También existen otras funciones y definiciones en la librería, que tu profesor de taller te debe explicar. La idea es que simplemente aprendas a utilizar las funciones mencionadas a tu favor para desarrollar esta tarea.

#### IV. ACTIVIDADES DE LOS TALLERES Y AUTÓNOMO.

En la **primera clase de taller** asociado a esta tarea desarrollarás junto a tu profesor las siguientes funcionalidades:

- Imprimir el arco de mayor costo y su costo.
- Imprimir los adyacentes y grado para cada nodo del grafo.
- Imprimir el recorrido en profundidad desde cada nodo.

En la **segunda clase de taller** asociado a esta tarea desarrollarás junto a tu profesor las siguientes funcionalidades:

- Imprimir si se trata de un grafo  $k$ -regular o no. Si es  $k$ -regular debes imprimir el valor de  $k$ .
- Imprimir si se trata de un grafo conexo o desconexo. Si es desconexo debe indicar además el número de componentes conexas.
- Imprimir los caminos de costo mínimo desde cada nodo (usando Dijkstra) a todos los otros. Si no existe un camino el programa debe imprimir un mensaje.

#### Autoestudio:

Para tu autoestudio deberás desarrollar el resto de las funciones que no se realicen en taller. Es decir:

- Imprimir el recorrido en amplitud desde cada nodo.
- Generar e imprimir el grafo complemento.
- Imprimir si se trata de un grafo completo o no
- Imprimir los sucesores y grado de salida para cada nodo del grafo.
- Imprimir los antecesores y grado de entrada para cada nodo del grafo.
- Imprimir el recorrido en amplitud desde cada nodo.

**Recuerda contactar a tu profesor de taller por cualquier duda que tengas en este proceso.**

#### IV. SOBRE BUENAS PRÁCTICAS DE PROGRAMACIÓN.

1. Debes usar identificadores representativos para tus constantes, variables, parámetros de entrada y funciones.
2. Las variables locales a cada función **debes** definir las al **principio** de la función. Puedes darle valores iniciales al momento de definir las (esto se llama “inicializarlas”).
3. Tu código debe estar correctamente *indentado* (uso de sangrías para cada sub-bloque de instrucciones), esto incluye el correcto alineamiento de las llaves (“{” y “}”) que delimitan tales bloques.
4. Tu código no puede presentar más de 1 línea en blanco.
5. Tu código no puede poseer instrucciones “basura”.



6. Debes comentar cada una de las funciones que definas como se indicó en la tarea 0.

#### IV. SOBRE LA EVALUACIÓN.

En el taller que **corresponda la evaluación de esta tarea** (tercer taller luego de iniciada la Tarea N°2) se pedirá que escribas en **papel** solo una función que realice alguna tarea sobre el arreglo que aprendiste a manejar en los talleres previos.