



GRADO

## PROGRAMACIÓN Y ESTRUCTURAS DE DATOS AVANZADAS

ENUNCIADO PRÁCTICA 2

Curso 2021-2022

## 1.- ENUNCIADO DE LA PRÁCTICA

Se trata de calcular el camino de coste mínimo entre cada par de nodos de un grafo dirigido utilizando el algoritmo de Floyd, algoritmo que utiliza el esquema de Programación Dinámica. El resultado del algoritmo será, para cada par de nodos, la secuencia de nodos del camino más corto y su valor total o longitud.

El enunciado del problema, la descripción de la solución, junto con un ejemplo explicativo se puede encontrar en el apartado 5.7 del texto base de la asignatura.

## 2.- REALIZACIÓN DE LA PRÁCTICA

### 2.1.- Diseño del algoritmo

La práctica constará de una memoria y de un programa en java original que resuelva el problema aplicando el esquema indicado.

### 2.2.- Argumentos y parámetros

La práctica se invoca usando la siguiente sintaxis:

```
java floyd [-t][-h] [fichero entrada] [fichero salida]
```

o

```
java -jar floyd.jar [-t][-h] [fichero entrada] [fichero salida]
```

Los argumentos son los siguientes:

- **-t:** traza cada invocación recursiva de manera que se describa la parametrización de cada llamada recursiva.
- **-h:** muestra una ayuda y la sintaxis del comando.
- **fichero\_entrada:** es el nombre del fichero del que se leen los datos, en este caso la representación del grafo.
- **fichero\_salida:** es el nombre del fichero que se creará para almacenar la salida. Si el fichero ya existe, el comando dará un error. Si falta este argumento, el programa muestra el resultado por pantalla.

Por ejemplo:

```
$ java floyd -h <ENTER>
```

```
SINTAXIS: floyd [-t][-h] [fichero entrada] [fichero salida]
          -t                      Traza el algoritmo
          -h                      Muestra esta ayuda
          [fichero entrada]       Matriz de adyacencia que representa el grafo
          [fichero salida]        Para cada par de nodos: la lista de nodos del
                                   camino más corto y su valor o longitud
```

### 2.3- Datos de entrada

El formato del fichero de entrada correspondiente a un grafo dirigido de  $n$  nodos será de  $n$  filas y cada una contendrá  $n$  valores separados por espacios. Los valores vacíos se denominarán con un guión (-). Estas  $n$  filas x  $n$  columnas representan el grafo mediante una matriz de adyacencia,  $M$ , en la que el valor  $M[i,j]$  representa el valor de la arista entre los nodos  $i$  y  $j$ .

En caso de que el fichero de entrada no exista, leerá los datos por la entrada estándar con similar formato.

### 2.4- Datos de salida

La salida consistirá para cada par de nodos en:

- Su identificación.
- La lista de nodos del camino más corto y su valor o longitud.

Por ejemplo:

```
[1, 2]: 3,4,5: 45
...
```

Expresando que el camino más corto entre los nodos 1 y 2 se obtiene pasando por el 3, el 4 y el 5, en ese orden, con un valor, coste o longitud total de 45.

En caso de que el fichero de salida no se indique en la llamada al programa, se escribirá el resultado por la salida estándar.

### 2.5.- Implementación del algoritmo

El programa se desarrollará en Java siguiendo un diseño orientado a objetos. Los detalles del entorno recomendado se encuentran en la guía de la asignatura. Se valorará el diseño OO y la eficiencia del desarrollo.

### 3.- CUESTIONES TEÓRICAS DE LA PRÁCTICA

- 1) Indica y razona sobre el coste temporal y espacial del algoritmo.
- 2) Explica qué otros esquemas pueden resolver el problema y razona sobre su idoneidad.

### 4.- ENTREGA DE LA PRÁCTICA

#### 4.1 Material que hay que entregar al Tutor

Se confeccionará una memoria, en PDF con el siguiente índice:

1. **Portada** de la memoria con **nombre, apellidos, dni y dirección de correo**.
2. Respuesta a las cuestiones teóricas planteadas en este enunciado.
3. Un ejemplo de ejecución para distintos tamaños del problema.
4. Un listado del código fuente completo.

#### 4.2 Normativa de las prácticas en relación con el Centro Asociado:

1. La asistencia a las sesiones de prácticas es obligatoria.
2. El calendario y procedimiento para asistir a las sesiones de prácticas **está publicado en su Centro Asociado o bien aparece en el foro correspondiente a su centro en el curso virtual**.
3. El plazo de entrega de la documentación y de la práctica **lo establece el Tutor de prácticas de cada Centro Asociado o Campus**.
4. El Tutor califica la práctica, informa al alumno y en su caso la revisa de acuerdo con los horarios y procedimiento que establezca el Centro Asociado.
5. Todos los alumnos deberán registrarse a través del Curso Virtual en el grupo del Tutor/a con el que hayan asistido a las sesiones presenciales obligatorias a fin de que su práctica pueda ser calificada.
6. La práctica se debe aprobar en la misma o anterior convocatoria para que se pueda calificar la asignatura. En caso contrario la calificación será de suspenso.
7. La práctica se entregará tanto en el entorno virtual como al Tutor. La falta de cualquiera de ellas será motivo suficiente para quedar excluida de la convocatoria.

El alumno debe asegurarse de que no se da ninguna de las siguientes circunstancias, ya que implican automáticamente una calificación de suspenso:

- **Código:** el código no compila, no está desarrollado en Java, no se corresponde con el pseudocódigo recogido en la documentación, no es original, está copiado de la red, academia, compañero, etc., o no sigue un diseño OO encapsulado o modular.
- **Ejecutable:** el ejecutable no termina, se queda sin memoria con ejemplares pequeños o aborta sin justificación. El ejecutable no lee los ficheros previstos en el formato adecuado. No trata los argumentos o no se ajusta a las especificaciones.
- **Documentación:** No se presenta en el soporte indicado por el tutor o está incompleta.
- **Soporte:** No se puede leer, o contiene un virus de cualquier tipo. A este respecto, las prácticas en las que se detecte cualquier tipo de virus estarán suspensas.

Los **alumnos estudiando en el EXTRANJERO** se deberán poner en contacto con el profesor tutor que se indicará en los foros.