

# Camino de costo mínimo desde un vértice fijo

## 1. Introducción

El objetivo del laboratorio es la implementación de varios algoritmos de caminos de costo mínimo, desde un vértice fuente fijo. En específico, debe implementar: (I) el algoritmo de Bellman-Ford, (II) el algoritmo de costo mínimo sobre DAG, (III) el algoritmo de Dijkstra, y (IV) el algoritmo para obtener el camino crítico en un PERT <sup>1</sup> *chart*.

## 2. Actividad a realizar

Se quiere agregar a la librería *libGrafoKt* las siguientes clases:

**CCM\_BellmanFord:** Obtiene el camino de costo mínimo desde un vértice fuente dado, hasta cada uno de los vértices de un digrafo, usando el algoritmo de Bellman Ford.

**CCM\_DAG:** Obtiene el camino de costo mínimo desde un vértice fuente dado, hasta cada uno de los vértices de un digrafo acíclico DAG.

**CCM\_Dijkstra:** Obtiene el camino de costo mínimo desde un vértice fuente dado, hasta cada uno de los vértices de un digrafo, usando el algoritmo de Dijkstra.

**CaminoCriticoPERT:** Obtiene el camino crítico de un PERT *chart*, modelado como un digrafo acíclico.

Se le proporcionará de un código base, contenido en el archivo `codigoSemana7.tar.xz`. Este código contiene los archivos adicionales que usted debe agregar a la librería *libGrafoKt*. Se debe completar y documentar el código de las actividades a realizar. Puede hacer uso de las clases de la librería de Kotlin para su implementación. Cada una de las operaciones en las clases tiene una breve descripción la misma. Esa descripción debe ser borrada de su código de entrega. En su lugar deben colocar para cada una documentación de las operaciones en las que se indique *descripción*, *parámetros*, *precondiciones*, *postcondiciones* y *tiempo de la operación*. El tiempo de las operaciones debe ser dado en número de lados y/o número de vértices, cuando eso sea posible. Sus implementaciones deben ser razonablemente eficientes. Debe entregar la librería *libGrafoKt* completa, con todos los códigos de las implementaciones de esta semana y de las semanas anteriores, junto con un archivo *makefile*, que compile solamente a la librería. Las implementaciones de sus soluciones deben estar basadas en los pseudo códigos de los algoritmos vistos en clase y en [1].

---

<sup>1</sup>PERT es acrónimo de *Program Evaluation and Review Technique*.

### 3. Condiciones de entrega

Los códigos del laboratorio y la declaración de autenticidad debidamente firmada, deben estar contenidos en un archivo comprimido, con formato *tar.xz*, llamado *LabSem7-X-Y.tar.xz*, donde *X* y *Y* son los números de carné de los estudiantes. La entrega de este archivo debe hacerse por medio de la plataforma *Classroom* antes de las 11:59 P.M. del día domingo 10 de julio de 2022.

### Referencias

- [1] CORMEN, T., LEIRSESON, C., RIVEST, R., AND STEIN, C. *Introduction to Algorithms*, 3ra ed. McGraw Hill, 2009.