UNIDAD TEMÁTICA 4 – GRAFOS DIRIGIDOS

PRACTICOS DOMICILIARIOS INDIVIDUALES - 5

Utilizando el código fuente ya desarrollado para el TDA GRAFO DIRIGIDO, implementar las siguientes funcionalidades.

Ejercicio 1 – CICLOS

Una gran cantidad de aplicaciones de grafos dependen de que los mismos no contengan ciclos, por lo que determinar esto con la mayor eficiencia es bien importante.

Se desea entonces desarrollar una funcionalidad para **Determinar si el Grafo contiene ciclos** (al menos uno!):

- Desarrolla el algoritmo en seudocódigo y analiza el orden del tiempo de ejecución. ¿Puedes encontrar un enfoque mejor?

Ejercicio 2 - Ordenación topológica

El "sort topológico" es una operación práctica de grafos muy utilizada en una gran variedad de problemas de aplicación. Para incrementar la comprensión sobre esta importante funcionalidad, empieza revisando el libro de **AHO**, Capítulo 6, Sección 6.6. Luego te recomendamos leer cuidadosamente el libro de **Skiena**, Secciones 5.10.1 y 15.2.

Se desea entonces: dado un Grafo Dirigido Acíclico, desarrollar una funcionalidad que emita una ordenación topológica del mismo.

- Desarrolla el algoritmo en seudocódigo y analizar el orden del tiempo de ejecución del mismo. Especifica claramente las precondiciones y postcondiciones para esta funcionalidad.
- Implementa la funcionalidad en el TDA Grafo Dirigido (y TVertice).
- ¿Cómo harías para obtener ***todas las ordenaciones topológicas existentes?***

Ejercicio 3

Encontrar los "componentes fuertes" o "componentes conexos" de un grafo es una tarea que se encuentra en el núcleo de muchas aplicaciones de grafos.

Probar si un grafo es conexo es un paso esencial de pre-procesamiento para cualquier algoritmo de grafos. A menudo se tienen errores sutiles, difíciles de encontrar, cuando un algoritmo se ejecuta sólo sobre un componente de un grafo desconectado. Las pruebas de conectividad son tan rápidas y fáciles que siempre deberíamos verificar la integridad de nuestro grafo de entrada.

Deseamos en primer lugar diseñar **un algoritmo que verifique si un grafo dirigido de entrada es fuertemente conexo**.

- Ve el ejercicio de libro de **Weiss, 14.21,** y el libro de **Skiena, sección 5.10.2.**
- Desarrolla en seudocódigo una funcionalidad para determinar si el grafo de entrada está fuertemente conectado. Analiza el orden del tiempo de ejecución.
- Implementa una operación a nivel del TDA Grafo Dirigido, "esConexo".

Luego también deseamos diseñar un algoritmo para, dado un grafo dirigido de entrada, **hallar todos sus componentes fuertes.**

Para completar este ejercicio, te recomendamos:

1. Comienza revisando la lectura del libro de AHO, Capítulo 6, Sección 6.7, y pensando cómo implementar el algoritmo con las estructuras de datos y operaciones que ya tenemos.
2. Sigue con el libro de SKIENA, Sección 5.10.2 y luego la sección 15.1
3. Desarrolla un algoritmo en seudocódigo para hallar los Componentes Conexos del grafo. Analiza el orden del tiempo de ejecución del mismo
4. Implementa la funcionalidad (a nivel de TGrafoDirigido y lo que sea necesario a nivel de TVertice) que, dado un Grafo Dirigido, obtenga sus componentes fuertes. ¿Cómo sería representada esta salida?