Programación con Memoria Dinámica

Contenedor DGraph

I. Objetivo de la actividad

La evaluación extra-aúlica del parcial, en la materia de Programación con Memoria Dinámica (PMD), tiene por objetivo medir su capacidad de resolución de problemas computacionales, aplicando lo aprendido en algunos o todos los temas que se han abordado en el curso; además, se pretende evidenciar su aprendizaje en programación utilizando el lenguaje C, así como sus habilidades de investigación y capacidad del trabajo en equipo.

II. Descripción del problema

En esta actividad deberás implementar un contenedor que, permita al usuario crear grafos dirigidos que almacene datos genéricos.

2.1 Contenedor DGraph

Un contendor DGraph es un tipo de dato abstracto que implementa el concepto de grafo dirigido. Un grafo dirigido consta de un conjunto finito de vértices **V** (también llamados nodos) y un conjunto aristas **E** (que representan pares ordenados de vértices). Un grafo dirigido se puede representar mediante un diagrama de la siguiente forma:

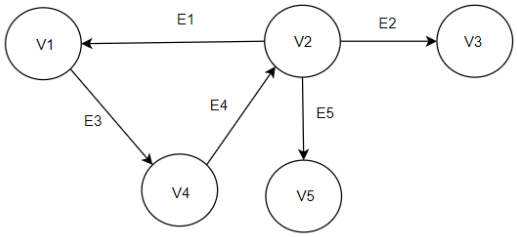

Figura 1 Representación gráfica de un grafo dirigido

En un grafo dirigido cada vértice $\mathbf{V_x}$ (nodo) puede tener **N** aristas E_x (enlaces). Adicionalmente, en un grafo cada arista E_x puede tener asociado algún valor, como una etiqueta simbólica o un atributo numérico (costo, capacidad, longitud, etc.). Veamos el ejemplo de grafo dirigido en el que cada vértice representa una ciudad y las aristas tienen asociadas un valor que representa la distancia que existe entre las ciudades vecinas.

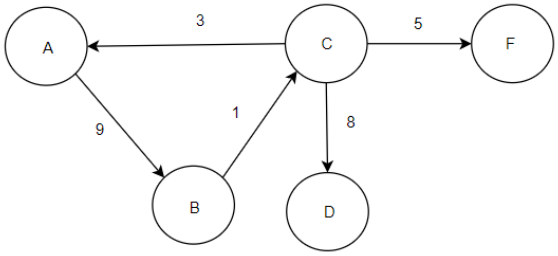

Figura 2 Ejemplo de un grafo dirigido con pesos en las aristas

Otro concepto relevante al trabajar con grafos es el término de vecindad. En el grafo de la figura 2, se tienen las siguientes vecindades:

- Los vértices A, D y F **son vecinos de**l vértice C. Porque **existen aristas que van del vértice C a los vértices** A, D y F.
- El vértice B es vecino del vértice A.
- El vértice C es vecino del vértice B.

En general podemos decir que un vértice ***y*** es vecino del vértice ***x***, si existe una arista que va del vértice ***x*** al vértice ***y***.

Ahora que conocemos qué es un grafo dirigido, veamos el tipo de operaciones que se pueden realizar sobre un grafo dirigido.

2.1 Operaciones básicas sobre un grafo dirigido

Un contendor DGraph normalmente provee las siguientes funciones básicas:

a)	*createDGraph()*	Crea una nueva instancia de un contenedor DGraph.
b)	*sizeGraph(DG)*	Retorna el número de vértices que existen en el grafo.
c)	*adjacent(DG, x, y)*	Retornar true si en el contendor **DG**, existe una arista entre los vértices: **x** y **y**.
d)	*neighbors(DG, x)*	Retorna los datos que se encuentran en los vértices ***y*** que son vecinos del vértice **x**. Un vértice ***y*** es vecino de ***x***, si existe una arista de ***x*** a ***y***.
e)	*addVertex(DG, x)*	Agrega un nuevo vértice ***x***, si no existe en el grafo ***DG***.
f)	*removeVertex(DG, x)*	Remueve el vértice ***x***, si existe en el grafo ***DG***.
g)	*addEdge(DG, x, y, z)*	Agrega al grafo ***DG*** una arista ***z*** del vértice ***x*** al vértice ***y***, si no existe.

h)	*removeEdge(DG, x, y)*	*R*emueve del grafo ***DG*** la arista del vértice ***x*** al vértice ***y***, si existe.
i)	*getVertexData(DG, x)*	Retorna el dato almacenado en el vértice ***x***.
j)	*setVertexData(DG, x, d)*	Almacena el dato ***d*** en el vértice ***x***.
k)	*getEdgeLabel(DG, x, y)*	*R*etorna la etiqueta asociada con la arista que conecta a los vértices: ***x***, ***y***;
l)	*setEdgeLabel(DG, x, y, l)*	Asigna la etiqueta l a la arista asociada con la arista que une a los vértices: ***x***, ***y***.
m)	*destroyDGraph(DG)*	Destruye el contenedor **DG**, liberando toda la memoria dinámica

2.2 Requisitos

Para el diseño y posterior implementación del contenedor, es necesario que consideres los siguientes requerimientos funcionales:

a) El contenedor DGraph debe ser implementado como un tipo de dato abstracto que permita almacenar datos genéricos.

b) La interfaz del contenedor deberá ser definida en un archivo llamado DGraph.h

c) La implementación del contenedor deberá codificarse en un archivo llamado DGraph.h

d) Las pruebas realizadas para verificar el buen funcionamiento del contenedor DGraph deberá codificarse en el archivo llamado testDGraph.c

3. Evaluación

Los productos que debe entregar en CANVAS son:

- Código fuente del contenedor: DGraph.h, DGraph.c, testDGraph.c
- Reporte de la actividad (formato libre)

Los criterios para considerar en la evaluación de la actividad son los siguientes:

Requerimientos funcionales (80%)

Funcionalidades	10	5	2	0
createDGraph	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada
sizeDGraph	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada
adjacent	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada
neighbors	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada
addVertex	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada
removeVertex	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada
addEdge	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada
removeEdge	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada
getVertexData	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada
setVertexData	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada
getEdgeLabel	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada
setEdgeLabel	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada
destroyDGraph	Funciona de manera correcta	Tiene errores de sintaxis	Tiene errores de lógica	No fue implementada

Nota: para tener derecho a calificación, es requisito indispensable que tu contenedor sea un tipo de dato abstracto, almacene para datos genéricos y utilice memoria dinámica.

Reporte (20%)

Criterios	10	5	0
Portada	Incluye los elementos necesarios para identificar la asignatura, el trabajo realizado, la institución educativa, los integrantes del equipo y fecha. Adicionalmente, tiene una buena presentación.	Incluye los elementos necesarios para identificar la asignatura, el trabajo realizado, la institución educativa, los integrantes del equipo y fecha. Sin embargo, tiene una mala presentación.	No se incluye o no cumple con los requisitos para una calificación mayor
Introducción	Brinda un resumen de su trabajo que permite conocer al lector de manera **breve** cuál es el contenido de su trabajo: problema, propuesta de solución y resultados.	Brinda un resumen de su trabajo, pero no permite conocer al lector de manera breve cuál es el contenido de su trabajo: problema, propuesta de solución y resultados.	No se incluye o no cumple con los requisitos para una calificación mayor
Análisis del problema	Brinda evidencia del análisis realizado sobre el problema: incluye redacción y diagramas.	Brinda evidencia del análisis realizado sobre el problema: incluye redacción, pero no incluye diagramas.	No se incluye o no cumple con los requisitos para una calificación mayor
Diseño de la solución	Brinda evidencia del diseño de la solución que se realizó en base a un análisis previo del problema: incluye redacción y diagramas.	Brinda evidencia del diseño de la solución que se realizó en base a un análisis previo del problema: incluye redacción y, pero no incluye diagramas.	No se incluye o no cumple con los requisitos para una calificación mayor
Pruebas	Muestra evidencia de las pruebas que fueron realizadas para validar cada requerimiento funcional.	Muestra evidencia de las pruebas que fueron realizadas, pero no son suficientes para validar cada requerimiento funcional.	No se incluye
Conclusiones	Brinda evidencia de una reflexión y discusión realizada entre los autores del trabajo respecto a: problemas, aprendizajes, resultados y oportunidades de mejora.	Brinda evidencia de una reflexión y discusión realizada entre los autores del trabajo, sin embargo, no aborda alguno de los siguientes aspectos: problemas, aprendizajes, resultados o oportunidades de mejora.	No se incluye