



Algoritmos y Estructuras de Datos II

Trabajo Práctico 4 y 5 - Ejercicios adicionales

OBJETIVOS

- Repasar y fortalecer el concepto de grafos mediante su implementación con arrays.
- Repasar y fortalecer el concepto de Tipo de Dato Abstracto (TDA) mediante su implementación y uso en diferentes estructuras de datos.

EJERCICIOS PROPUESTOS

1. Codificar un tipo de dato *tGrafoPlayas* para modelar un grafo ponderado de hasta 15 vértices, que representan las playas de la costa atlántica Argentina. El peso de los arcos es un dato compuesto que representa la distancia en km y la cantidad promedio de combustible en litros que se consume entre nodos. Utilizar la implementación de grafos con *arrays*.

Además:

- Codificar una función que permita agregar un vértice al grafo definido anteriormente.
- Codificar la función que permita agregar un arco al grafo definido anteriormente.
- Codificar una función que permita imprimir la matriz de pesos ó la matriz de adyacencia, según sea un valor del tipo *char* que se recibe como parámetro (a-adyacencia p-pesos). En el caso de la matriz de pesos, los valores de los arcos deberán mostrarse como en el siguiente ejemplo: (90.24 km, 35.25 ltrs)

2. Codificar tipo de dato *tGrafoDiscord* para modelar un grafo ponderado de hasta 50000 vértices, que representan usuarios de *Discord*. El peso de los arcos representa la cantidad de servidores en común entre los nodos. Utilizar la implementación de grafos con *arrays*.

Además:

- Codificar la función que permita agregar un arco al grafo definido anteriormente.
- Codificar la función que permita quitar un arco al grafo definido anteriormente.
- Codificar la función que permita visualizar la matriz de adyacencia del grafo.

3. Crear un Tipo de Dato Abstracto (TDA) llamado *tGrafoProyectos* para modelar un grafo ponderado con capacidad para hasta 200 nodos, representando distintos proyectos interconectados en una red de colaboración científica. Cada arista contendrá información sobre la distancia en kilómetros entre los centros de investigación asociados a los proyectos y el número de colaboradores involucrados. La

implementación del grafo se llevará a cabo utilizando una estructura basada en *arrays*. Además, se solicita:

- Desarrollar métodos para añadir y eliminar aristas en el grafo definido.
 - Agregar una función permita imprimir la matriz de adyacencia o la matriz de pesos, en función de un dato que se reciba como parámetro que indique cuál matriz se desea imprimir.
 - Escribir un programa que haga uso de este TDA, declare una variable grafo del tipo *tGrafoProyectos* y utilice las funciones disponibles para gestionar la colaboración científica entre los proyectos de investigación.
4. Desarrollar un Tipo de Dato Abstracto (TDA) llamado *tGrafoHoteles* para modelar un grafo ponderado con capacidad para hasta 120 vértices, representando hoteles conectados en una cadena hotelera. Cada arista contendrá información sobre la distancia en kilómetros entre hoteles y el nivel de ocupación promedio. La implementación del grafo se realizará utilizando *arrays*. Además, se solicita:
- Codificar funciones para agregar y quitar aristas en el grafo definido.
 - Crear un programa que haga uso de este TDA, declare una variable grafo del tipo *tGrafoHoteles* y utilice las funciones disponibles para gestionar la cadena hotelera.
5. Una librería requiere de un programa para gestionar los libros disponibles. Se necesita un TDA para implementar una lista enlazada que gestione los datos de los libros disponibles en la librería. Cada libro tendrá los siguientes datos: título, autor, género y cantidad disponible. Se pide crear un tipo de dato abstracto (TDA) para la lista enlazada del problema, que incluya funciones para:
- Crear lista, preguntar si está vacía
 - Insertar un nuevo libro (tanto al principio como en una posición específica)
 - Eliminar un libro del principio de la lista
 - Mostrar la cantidad disponible de libros por género

Además, escribir un programa para probar las operaciones del TDA que contenga funciones para: Registrar libros en la lista, eliminar un libro del inventario, y mostrar cuántos libros de un determinado género están disponibles, según el género seleccionado por el usuario, siguiendo un formato similar al siguiente ejemplo:

*** Cantidad de libros disponibles por género ***

Género	Cantidad

Ciencia Ficción:	15