

UNIVERSIDAD DEL VALLE DE GUATEMALA

CC 2016 -Estructura de Datos

Sección 10

Ing.Michaëlle Alexander Pérez Riz



Hoja de Trabajo #10

Jonathan Zacarías - 231104

Guatemala, 21 de Mayo de 2024

Análisis:

- Requerimiento
 - ¿Qué acciones debe poder hacer su programa?
 - Leer datos del grafo desde un archivo:
 - El programa debe ser capaz de leer un archivo de texto (guategrafo.txt) que contiene la información de las ciudades y las distancias entre ellas para construir el grafo.
 - Consultar la ruta más corta entre dos ciudades:
 - El usuario debe poder ingresar dos nombres de ciudades y el programa debe calcular y mostrar la distancia más corta entre ellas, así como la ruta detallada.
 - Calcular y mostrar el centro del grafo:
 - El programa debe determinar cuál es la ciudad que sirve como centro del grafo, es decir, la ciudad con la menor distancia máxima a todas las demás ciudades.
 - Modificar el grafo:
 - Permitir al usuario realizar modificaciones en el grafo como interrumpir el tráfico entre ciudades o establecer nuevas conexiones con sus respectivas distancias.
 - Mostrar la matriz de adyacencia del grafo:
 - Debe ser capaz de mostrar la matriz de adyacencia actualizada después de cualquier modificación.
 - ¿Con qué datos va a trabajar?
 - Archivo de Entrada (guategrafo.txt):
 - Contiene líneas con el nombre de dos ciudades y la distancia entre ellas en kilómetros. Este archivo es la fuente de datos inicial para construir el grafo.
 - Descripción de clases:Métodos y Atributos de clases:
 - Clase Grafo:
 - Atributos:
matrizAdyacencia: Matriz que representa el grafo con las distancias entre los nodos.
 - Métodos:
Grafo(int numVertices): Constructor que inicializa la matriz de adyacencia con el número de vértices.

agregarArco(int origen, int destino, int peso): Método para añadir un arco (conexión) entre dos nodos con un peso (distancia).

eliminarArco(int origen, int destino): Método para eliminar un arco entre dos nodos.

obtenerMatrizAdyacencia(): Método para obtener la matriz de adyacencia actual.

■ Clase FloydWarshall:

● Métodos Estáticos:

ejecutarFloydWarshall(int[][] matrizAdyacencia): Aplica el algoritmo de Floyd-Warshall para calcular las distancias más cortas entre todos los pares de nodos.

obtenerRuta(int[][] matrizRutas, int origen, int destino):

Obtiene la ruta más corta entre dos nodos dados utilizando la matriz de rutas generada por Floyd-Warshall.

■ Clase CentroGrafo:

● Métodos Estáticos:

calcularCentro(int[][] distancias): Calcula la ciudad que sirve como centro del grafo, basado en las distancias calculadas por Floyd-Warshall.

■ Clase (Main):

● Métodos:

main(String[] args): Método principal que inicia el programa, maneja la interacción con el usuario y coordina la ejecución de las otras clases y métodos.

Diagramas

Clases

