Dada la consigna de la actividad 2, modificar o rehacer el programa, para que se pida origen y destino y se imprima el camino más corto entre dos vértices.

import java.util.Arrays;

import java.util.Scanner;

public class OptimizacionRutas {

static final int INF = 99999; // Representa infinito para rutas inexistentes

public static void floydWarshall(int[][] tiempos, int numCentros, int[][] siguiente) {

int[][] distancias = new int[numCentros][numCentros];

// Inicializamos la matriz de distancias y la matriz de rutas

for (int i = 0; i < numCentros; i++) {

for (int j = 0; j < numCentros; j++) {

distancias[i][j] = tiempos[i][j];

if (tiempos[i][j] != INF && i != j) {

siguiente[i][j] = j; // Si hay conexión, establecemos el siguiente nodo

} else {

siguiente[i][j] = -1; // Si no hay conexión, ponemos -1

}

}

}

// Aplicar el algoritmo de Floyd-Warshall

for (int k = 0; k < numCentros; k++) {

for (int i = 0; i < numCentros; i++) {

for (int j = 0; j < numCentros; j++) {

if (distancias[i][k] != INF && distancias[k][j] != INF &&

distancias[i][k] + distancias[k][j] < distancias[i][j]) {

distancias[i][j] = distancias[i][k] + distancias[k][j];

siguiente[i][j] = siguiente[i][k]; // Actualizar el camino

}

}

}

}

// Detectar ciclos negativos

for (int i = 0; i < numCentros; i++) {

if (distancias[i][i] < 0) {

System.out.println("⚠️ Se ha detectado un ciclo negativo en el centro " + (i + 1));

return;

}

}

// Imprimir la matriz de distancias mínimas

System.out.println("\n🚛 Tiempos mínimos de entrega entre centros:");

imprimirMatriz(distancias, numCentros);

// Pedir origen y destino

Scanner scanner = new Scanner(System.in);

System.out.print("\nIngrese el centro de origen (1-" + numCentros + "): ");

int origen = scanner.nextInt() - 1;

System.out.print("Ingrese el centro de destino (1-" + numCentros + "): ");

int destino = scanner.nextInt() - 1;

// Mostrar camino más corto

if (siguiente[origen][destino] == -1) {

System.out.println("No hay ruta disponible entre los centros seleccionados.");

} else {

System.out.println("\n⏳ Tiempo mínimo: " + distancias[origen][destino] + " minutos");

System.out.print("Ruta óptima: ");

imprimirCamino(origen, destino, siguiente);

System.out.println();

}

scanner.close();

}

public static void imprimirMatriz(int[][] matriz, int numCentros) {

for (int i = 0; i < numCentros; i++) {

for (int j = 0; j < numCentros; j++) {

if (matriz[i][j] == INF) {

System.out.print("INF\t");

} else {

System.out.print(matriz[i][j] + "\t");

}

}

System.out.println();

}