

Reporte de Tarea 3

Liqing Yosery Zheng Lu, C38680, liqing.zheng@ucr.ac.cr

Resumen—El Sistema Nacional de Tipificación de Emergencias Potenciales (SiNTiEmPo) administra una red de centros de atención distribuidos en distintas ciudades. Estas ciudades están conectadas por rutas cuyo tiempo promedio de viaje es conocido. En este proyecto se modela dicha red como un grafo dirigido ponderado y se implementa un sistema capaz de responder diversas consultas estratégicas, tales como la ubicación óptima para almacenar suministros, las rutas más cortas entre centros, y el ordenamiento de ciudades según eficiencia de conexión. Se hace uso del algoritmo de Floyd-Warshall para calcular todos los caminos más cortos entre pares de ciudades y se justifica su selección por su eficiencia y aplicabilidad a grafos densos con pesos positivos.

Palabras clave—grafos dirigidos, caminos más cortos, Floyd-Warshall, planificación de rutas, algoritmos de grafos

I. INTRODUCCIÓN

El Sistema Nacional de Tipificación de Emergencias Potenciales (SiNTiEmPo) mantiene centros especializados distribuidos en distintas ciudades del país, los cuales almacenan equipos y suministros necesarios para la atención de emergencias. Algunas de estas ciudades están conectadas entre sí por rutas directas, y para cada conexión se conoce el tiempo promedio de viaje entre las ciudades involucradas.

El objetivo principal de este proyecto es modelar dicha red de ciudades como un grafo dirigido ponderado, donde los vértices representan las ciudades y las aristas con peso representan los tiempos de viaje promedio entre ciudades conectadas directamente. A partir de este modelo, se desarrolla un sistema que permite responder eficientemente varias consultas fundamentales para la planificación y toma de decisiones dentro del SiNTiEmPo, tales como:

- Determinar la ciudad óptima para colocar mayor capacidad de equipo, minimizando el tiempo total de envío hacia las demás ciudades.
- Identificar desde qué ciudad debe despacharse equipo hacia una ciudad específica para que llegue en el menor tiempo posible.
- Encontrar las ciudades más cercanas o más lejanas entre sí, en términos de tiempo de viaje.
- Ordenar las ciudades según su tiempo promedio de viaje hacia las demás.

Para resolver estas consultas de manera eficiente, se utiliza el algoritmo de Floyd-Warshall, el cual permite obtener las distancias mínimas entre todos los pares de vértices en grafos con pesos positivos, como es el caso de esta red de transporte.

II. METODOLOGÍA

Al procesar un archivo CSV, el programa guarda los nombres de las ciudades en un arreglo y sus conexiones en una matriz de adyacencia, además de generar una matriz de predecesores para reconstruir caminos mínimos. Para optimizar

el tiempo de ejecución, se usa un enfoque de *programación dinámica*: cuando un archivo ya ha sido cargado y analizado, sus resultados se almacenan en memoria como una matriz de distancias. Así, si se vuelve a solicitar el mismo archivo, no es necesario recargarlo ni recalcular los valores, lo que reduce el tiempo de procesamiento y el uso de recursos, especialmente con archivos grandes o múltiples consultas.

Para calcular las distancias mínimas entre todos los pares de ciudades, se emplea el algoritmo de Floyd-Warshall, siguiendo la formulación de Cormen [1]. Este algoritmo analiza toda la matriz de adyacencia de una sola vez y obtiene todas las distancias mínimas necesarias para las cinco operaciones disponibles. Resulta adecuado para grafos ponderados con pesos positivos o negativos (siempre que no haya ciclos negativos) y su estructura matricial se adapta bien al almacenamiento en memoria para consultas posteriores.

Para determinar la ciudad óptima para almacenar mayor capacidad de equipo, se calcula la suma de tiempos mínimos de envío desde cada ciudad hacia todas las demás. La ciudad con la suma más baja es la opción más eficiente para centralizar el equipo.

Para identificar desde qué ciudad debe despacharse equipo hacia una ciudad específica para minimizar el tiempo de llegada, se revisan todas las ciudades de origen y se elige la que tenga la menor distancia mínima hacia la ciudad destino.

Para encontrar las ciudades más cercanas o más lejanas entre sí en términos de tiempo de viaje, se recorren todos los pares de ciudades (i, j) en la matriz de distancias mínimas (excluyendo los casos donde $i = j$). El par con la menor distancia corresponde a las más cercanas, mientras que el par con la mayor distancia indica las más lejanas.

Para ordenar las ciudades según su tiempo promedio de viaje hacia las demás, se calcula el promedio de las distancias mínimas desde cada ciudad a todas las otras. Luego, se ordenan en forma descendente según ese valor.

III. RESULTADOS

Los resultados completos de todas las consultas realizadas sobre los grafos representados en los tres archivos de prueba indicados en el enunciado se encuentran en un documento anexo. Este archivo en formato PDF está disponible en el repositorio, en el directorio `/tp3/doc`, con el nombre `Resultados.pdf`. Adicionalmente, se incluye la salida de terminal generada durante la recolección de resultados, la cual muestra la interacción del programa con el usuario. Dichos archivos se encuentran organizados en el directorio `/tp3/doc/outputs`, con un archivo de salida separado para cada archivo de prueba utilizado.

IV. MANUAL DE COMPILACIÓN

Para compilar el programa, siga los siguientes pasos:

1. Clonar el repositorio.
2. Abrir una terminal y navegar al directorio `tareas_programadas/tp3`.
3. Ejecutar el comando `make` en la terminal para compilar el programa. Esto generará el ejecutable correspondiente en la carpeta `bin`.
4. Para ejecutar el programa, utilizar el comando `./bin/main`.
5. Seguir las instrucciones que aparecen en pantalla para interactuar con el programa.

V. MANUAL DE USO

Al iniciar el programa, se presenta un menú principal donde se puede elegir el archivo que contiene el grafo a analizar. Para seleccionar una opción, basta con ingresar el número correspondiente en la consola.

- Opciones 1, 2 y 3 permiten elegir entre archivos de prueba predefinidos.
- La opción 4 permite cargar un grafo personalizado ingresando la ruta completa al archivo `.csv`.

Para salir del programa desde el menú principal, se puede ingresar la letra `X`.

Luego de seleccionar y procesar el archivo, se accede a un menú secundario con las diferentes acciones disponibles. Allí también se elige la opción deseada ingresando el número correspondiente.

En el caso de la **opción 2** (identificar desde qué ciudad debe despacharse equipo hacia una ciudad específica para que llegue en el menor tiempo posible), se debe ingresar la ciudad de destino. Puede seleccionarse por número o por nombre, sin distinción de mayúsculas o minúsculas.

Para regresar del menú secundario al menú principal, se puede ingresar la letra `X`.

REFERENCIAS

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009.



Liqing Yosery Zheng Lu Estudiante de computación en la Universidad de Costa Rica. Lleva el curso de Algoritmos y Estructuras de Datos durante el primer ciclo del año 2025 en la sede Rodrigo Facio.