

CI-0116 Análisis de Algoritmos y Estructuras de Datos
II ciclo 2024
Prof. Dr. Allan Berrocal Rojas

Tarea III

Índice

1	Indicaciones generales	2
2	Objetivo	2
3	Planteamiento del problema	2
4	Código Fuente de los Algoritmos	4
5	Pruebas que debe realizar	5
6	Reporte Escrito	5
7	Descripción del Producto a Entregar	6
8	Desglose de la evaluación	6



1. Indicaciones generales

Esta tarea es individual y debe entregarla en la fecha que se indica en la plataforma de Mediación Virtual. El(la) estudiante puede colaborar con sus compañeros(as) de manera ética. Es decir, antes de discutir los problemas de la tarea con sus compañeros(as), primero dedique al menos un día de trabajo individual y formule sus propias respuestas para la tarea. Aunque colabore con algún compañero(a), todo el material que desarrolla y entrega en su tarea debe ser hecho por usted mismo(a). Cada estudiante debe estar en capacidad de defender correctamente las respuestas de su trabajo de manera verbal si la persona docente o la persona asistente del curso formula una pregunta durante la revisión del trabajo. Si la persona docente sospecha que la tarea es parcial o totalmente producto de plagio o generada mediante un algoritmo generativo, la tarea recibirá una nota de cero y se iniciará el debido proceso estipulado en el [Reglamento de Orden y Disciplina de los Estudiantes Universidad de Costa Rica](#).

Debe hacer adecuada referencia a toda fuente externa de producción intelectual que utilizó para el desarrollo de este trabajo. Recuerde que se considera plagio presentar como propio, material parcial o totalmente creado por otras personas u obtenido de fuentes de información, como por ejemplo de libros, fuentes de Internet, sistemas de generación de texto basados en modelos de aprendizaje automático y profundo, entre otras.

2. Objetivo

El objetivo de la tarea es utilizar teoría de grafos y algoritmos asociados aplicados a la resolución de un problema específico. Se espera que la persona estudiante sea capaz de analizar el problema y diseñar una solución utilizando el conocimiento adquirido a partir del estudio de los algoritmos de grafos vistos en el curso.

3. Planteamiento del problema

Suponga que tiene una lista de ciudades en donde se encuentran centros de atención de emergencias de la Comisión Nacional de Emergencias en Costa Rica (CNE). En estos centros se realizan labores de planificación, se almacenamiento de equipos, materiales y suministros y se brinda atención de emergencias. Por ejemplo, sirven como albergues para personas que debido a una emergencia deben evacuar sus hogares. Algunas de las ciudades están conectadas con otras directamente mediante un camino. Para estas ciudades conectadas por un camino se tiene una tabla con el tiempo



promedio de viaje que se necesita para moverse de una ciudad a otra. Algunas ciudades no están conectadas directamente por un camino, pero se puede llegar a ellas pasando por una o más ciudades intermedias en el trayecto.

La información sobre las ciudades así como los tiempos de viaje promedio entre las ciudades que están conectadas por un camino se encuentra en un archivo de texto que usted puede leer.

Su misión es crear un programa que lea el archivo de entrada con la información de la CNE y procese dicha información para crear un grafo no dirigido $G = (V, E)$. En dicho grafo los vértices $v \in V$ representan las ciudades en donde se ubica un centro de atención de la CNE. Las aristas $(u, v) \in E$ del grafo conectan pares de ciudades y representan la existencia de una ruta directa entre ambos lugares. Las aristas tienen un peso w que representa un estimado del tiempo de viaje entre cada par de centros de atención. A modo de ejemplo, el grafo de la Figura 1 muestra 10 ciudades donde se encuentran centros de atención de emergencias de la CNE. El peso de las aristas representa en tiempo promedio de viaje entre cada par de ciudades que están conectadas directamente por un camino.

Además de crear un grafo con la información que lee del archivo, usted debe crear un programa que procese el grafo y extraiga cierta información útil para responder preguntas frecuentes del personal de la CNE para apoyar la toma de decisiones. Para ese propósito, su programa debe proveer un mecanismo básico de interacción que le permita al usuario consultar al menos las siguientes preguntas.

1. Indicar el nombre de la ciudad donde es más efectivo colocar mayor capacidad de equipo de manera que si se necesita repartir equipo ubicado en este centro hacia todos los demás, se tenga que invertir la menor cantidad de tiempo total de viajes. Si hay más de una ciudad que reúne esta condición, liste todos los nombres.
2. Dada una ciudad x , indicar el nombre de la ciudad y desde la cual es mejor despachar equipo de apoyo hacia la ciudad x para que llegue lo más pronto posible en caso de que ocurra una emergencia en la ciudad x . Si hay más de una ciudad que reúne esta condición, liste todos los nombres.
3. Indicar cuál es el par de ciudades (x, y) que están más distantes la una de la otra. Si hay más de una ciudad que reúne esta condición, liste todos los nombres.
4. Indicar cuál es el par (x, y) de ciudades que están menos distantes la una de la otra siempre que $x \neq y$. Si hay más de una ciudad que reúne esta condición, liste todos los nombres.
5. Producir una lista de todas las ciudades en orden creciente según el tiempo de viaje promedio desde esa ciudad hasta todas las demás ciudades.



Figura 1: Grafo de ejemplo con ciudades en los vértices y las aristas con peso representan el tiempo de viaje promedio entre cada par de ciudades.

4. Código Fuente de los Algoritmos

El código debe escribirse en el lenguaje de programación C++, y debe utilizar uno o más de los algoritmos de grafos estudiados en el curso.

Haga su solución modular, al menos distinga entre un módulo que lee el archivo de entrada, otro que procesa en grafo y extrae la información, y otro que le permite al usuario interactuar con el programa para responder cada una de las consultas a modo de menú.

Usted es libre de utilizar alguna interfaz gráfica si lo desea, o puede hacer que el programa



funcione mediante una terminal. En ambos casos la interacción debe ser intuitiva y simple.

Se debe usar el lenguaje en su versión estándar para evitar problemas de compilación. En particular, el código debe poder ser compilado usando `g++` y las bibliotecas estándar de C++. *Si el código no compila o no funciona correctamente, recibirá una nota de cero.*

Debe programar siguiendo las mejores prácticas estudiadas en la carrera incluyendo documentación de todas las clases y métodos que implementa. Para facilitar su labor, puede utilizar herramientas como [doxygen](#).

5. Pruebas que debe realizar

Antes de entregar su tarea utilice los archivos que se le brindan a modo de pruebas. Utilice el archivo `input_small.csv` para probar los resultados de su programa asegurándose de que produzca las respuestas correctas.

Posteriormente utilice el archivo `input_medium.csv` usted mismo (a) y haga todas las consultas que se listaron en la sección 3. Escriba los resultados que obtuvo para cada una de las preguntas con el archivo `input_medium.csv` y anote los resultados.

Haga la misma prueba para el archivo `input_large.csv` documentando el resultado que obtiene para cada una de las preguntas.

Documente los resultados obtenidos de las preguntas anteriores para cada caso en un archivo separado que puede tener capturas de pantalla de las salidas de su programa o listados de texto. Estos archivos los deberá entregar en **formato PDF únicamente**.

6. Reporte Escrito

Escriba un reporte breve de máximo tres páginas que indique al menos la siguiente información.

1. Su nombre completo y número de carné.
2. Breve planteamiento del problema a resolver.
3. Descripción de la solución que usted diseñó con una justificación clara de porqué eligió esa solución. Indique el nombre del algoritmo (o algoritmos) utilizado(s) en su solución y agregue referencias claras a su fuente.
4. Indique los resultados obtenidos para todas las preguntas en los grafos representados en los archivos `input_small.csv`, `input_medium.csv`, `input_large.csv`. Puede utilizar un forma-



to de tabla para los datos pequeños o hacer puede hacer referencia al nombre y sección del archivo anexo donde está dicho resultado.

5. Documentación clara sobre cómo compilar, ejecutar e interactuar con su programa.

El reporte se debe entregar en **formato PDF únicamente**. **Ningún otro formato será aceptado.**

7. Descripción del Producto a Entregar

La tarea tiene una única entrega que contiene el código fuente del programa y el reporte escrito en formato PDF siguiendo los lineamientos anteriores.

Entregue su trabajo en su repositorio privado de control de versiones en una estructura de directorios **exactamente** como se muestra a continuación: **[0.5 pts.]**

- `\tareas_programadas\tp3\`

Coloque aquí la versión del código fuente así como el documento de reporte en formato PDF.

Asegúrese de utilizar un archivo `.gitignore` para evitar subir al repositorio de control de versiones archivos temporales o autogenerados durante el proceso de compilación. **[0.5 pts.]**

Es responsabilidad de la persona estudiante verificar que la plataforma haya recibido la tarea y que esté intacta (bajando la tarea y verificando la integridad del comprimido). En caso de que haya múltiples entregas, se considerará solamente la *última*, y si esta se entregó después de la fecha y hora límites, se aplicará la penalización acordada en la CARTA AL ESTUDIANTE.

Si tiene dudas sobre cómo utilizar el repositorio de control de versiones, busque al asistente del curso o a la persona docente **con antelación** para recibir ayuda. No espere hasta el día de la entrega para solicitar ayuda porque corre el riesgo de que no se le pueda atender.

8. Desglose de la evaluación

El siguiente cuadro muestra un resumen de los elementos a evaluar en esta tarea.



Cuadro 1: Resumen de los elementos a evaluar

Elemento	Valor
Implementación del módulo que lee archivos de entrada y crea el grafo.	2
Implementación del algoritmo que procesa el grafo.	35
Implementación de la interfaz de usuario.	3
Implementación del programa que responde las consultas de usuario.	10
Datos obtenidos por el (la) estudiante al ejecutar su programa con los archivos datos.	5
Reporte escrito	10
Total	65 pts.