

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY



Escuela de Ingeniería

Análisis y diseño de algoritmos avanzados (Gpo 2)

Reflexión individual de la “Actividad Integradora 2”

Presenta:

Ana Cristina Munguía Romero, Matrícula A01740019

Guadalajara, Jal., 12 de noviembre de 2021

Reflexión individual de la “Actividad Integradora 2”

Esta actividad integradora consistía de 4 grandes partes: calcular la manera óptima de realizar un cableado de fibra óptica entre n colonias (utilizar la menor cantidad de cable posible), encontrar un camino viable para visitar todas las colonias en un solo recorrido, encontrar el mayor flujo de información posibles que se puede transmitir entre las colonias (desde la colonia de inicio 0 y la colonia final $n-1$), y finalmente, encontrar la región de Voronoi en la que se encuentra un punto dado (representando una nueva “contratación” que requiere conectarse a la central más cercana).

La entrega especificaba realizar la lectura de un archivo que contendría la información de *input* necesaria para realizar las distintas operaciones. Nosotros optamos por crear un archivo de texto (*.txt*) donde todos los datos meramente numéricos (la cantidad de nodos n y los valores de las matrices) se presentarían uno por línea, y los datos alfanuméricos (las coordenadas de las centrales al final del documento) se presentarían en el formato (x,y) . Para el último *input* del programa optamos por dejarlo abierto al usuario, tomándose directamente de una entrada a la terminal usando el mismo formato (x,y) .

Teniendo resuelta la lectura de archivos, empezamos con la implementación de los métodos para resolver las partes de la actividad. Para la primera parte usamos el algoritmo de Floyd-Warshall para encontrar la longitud del camino más corto entre cualesquiera dos nodos del grafo; este método terminó con una complejidad computacional cúbica de $O(n^3)$.

Para la segunda parte (encontrar un camino viable que recorra todas las colonias), optamos por hacer uso del algoritmo *Repetitive Nearest Neighbor*, que es un algoritmo avaro (y por ende no garantiza encontrar la solución óptima del problema) y tiene una complejidad también cúbica de $O(n^3)$, pues se trata de usar el método *Nearest Neighbor* (de complejidad cuadrada) y repetirlo n veces (cambiando el nodo de inicio) para seleccionar el mejor camino entre los obtenidos (de ahí que la complejidad sea cúbica).

En la tercera parte decidimos usar el algoritmo de Ford-Fulkerson para encontrar el flujo máximo del grafo, pero haciendo uso de la implementación particular de Edmonds-Karp, que encuentra si hay o no un *augmenting path* en el grafo residual haciendo uso de una búsqueda por amplitud (BFS). Esta implementación tuvo una complejidad de $O(EV^3)$, donde E es la cantidad de arcos en el grafo y V es la cantidad de nodos en el grafo.

La cuarta y última parte de la actividad integradora (que consistía en encontrar la región a la cual pertenecería cierto punto dado por el usuario) la resolvimos haciendo uso de la librería GCAL. Esta fue la parte más retadora para nosotros de la actividad, pues la librería es (a nuestro parecer) bastante complicada de utilizar; aún así, creemos que hubiera sido aún más complicado tratar de implementar el diagrama de Voronoi completamente desde cero. Pasamos muchos días aprendiendo a usar unos cuantos métodos de la librería y al final nos sentimos suficientemente cómodos con nuestra habilidad para ejecutarlos que optamos por editarlos a nuestra conveniencia. Aunque en la documentación de CGAL no se habla específicamente de la complejidad computacional de los métodos (solo se menciona que se debe ser cuidadoso al usarlas para evitar aumentarla accidentalmente), basándonos en las clases que hemos tenido, estimamos que la complejidad computacional no debería pasarse de $O(n \cdot \log(n))$.