Luis Cano Irigoyen – A00827178

Reflexión Individual
Actividad Integradora 2

Análisis y diseño de algoritmos avanzados, Gpo 8

En esta actividad integradora nos encontramos con una ciudad, formada por varias comunidades, a la cual se le quiere incursionar los servicios de Internet. Por ello requerimos utilizar diversos algoritmos trabajando con grafos y geometría para determinar las mejores soluciones a los diferentes puntos de la situación.

Para el primer punto tuvimos que determinar la forma óptima de cablear con fibra óptica para que se pueda compartir información entre cuales quiera dos colonias. Para ello utilizamos el algoritmo de Floyd Warshall para obtener un grafo óptimo. El grafo utilizado en este problema es un grafo ponderado, cuyos arcos cuentan con las distancias entre cada dos colonias. El algoritmo de Floyd Warshall obtiene todos los pares de aristas, o bien, colonias, y encuentra la menor distancia entre ellas. Este algoritmo cuenta con una complejidad de $O(N^3)$, N siendo la cantidad de Nodos o comunidades, dado a que tiene que buscar las menores distancias entre comunidades encontrado ya sea rutas directas de una arista a otra o rutas con otras aristas como intermediarias para alcanzar la distancia mínima entre el par, para ello utilizando 3 ciclos *for* anidados.

En el segundo punto fue necesario determinar la ruta más corta posible que visita todos los nodos una vez y regresa al punto inicial, por lo cual decidimos utilizar el algoritmo del *Travelling Salesman*. Para esta solución se comienza en una arista, y es necesario generar (N-1)! permutaciones del resto de las aristas, calculando las distancias de cada recorrido y guardando la mínima de todos las rutas realizadas. La complejidad para realizar cada una de estas permutaciones a todas las aristas termina siendo de O(N!). Una posible mejora para este punto sería el utilizar el algoritmo de Held-Karp para realizar el *Travelling Salesman*, ya que este algoritmo cuenta con una complejidad exponencial de $O(N^2\ 2^N)$, la cual es significativamente mejor que nuestra solución implementada. En Held-Karp se crean N-2 subsets del grafo, de tamaño N-1, y se realizan comparaciones recursivas hasta llegar a la ruta más corta. La desventaja que podría tener Held-Karp sería que su complejidad de espacio es $O(N2^N)$.

Para resolver el tercer punto determinamos la capacidad máxima de transmisión de datos entre colonias. Para ello utilizamos el algoritmo de Ford Fulkerson para obtener el flujo máximo desde el nodo inicial (A) al nodo final de un grafo. Este algoritmo busca el flujo máximo del recorrido al mismo tiempo que checa que el flujo entre dos aristas no exceda a la capacidad máxima de transmisión entre ellas. Para realizar la solución hicimos uso de una red residual, siendo esta el grafo que indica en dónde se puede admitir más flujo, en donde cada arco cuenta con una capacidad residual siendo esta su capacidad máxima menos el flujo que tiene actualmente. Luego, mientras existiera un camino aumentante en el grafo, fuimos aumentando el flujo a través del camino, guardando el valor del flujo máximo y actualizando la red residual. Originalmente contábamos con una solución de Ford Fulkerson la cual determinaba si había un camino aumentante, luego usando la capacidad residual determinaba el flujo máximo y por último actualizaba la red residual, en una complejidad total de $O(EN^3)$, siendo E la cantidad de "Edges" o "Arcos". En nuestra solución final, utilizamos la idea de la implementación del Edmonds-Karp algorithm, en donde al realizar el BFS al encontrar el camino aumentante uno de los arcos termina saturado, y podemos tomar su valor como el flujo del camino que vamos a sumar a nuestro flujo actual, reduciendo la cantidad de pasos y la complejidad de tiempo a tan solo $O(NE^2)$.

Por último, considerando las ubicaciones geográficas de N centrales a las cuales se conectan las casas, considerando que se hace una nueva contratación del servicio, tuvimos que determinar cual de estas centrales es la más cercana geográficamente a dicha nueva contratación. Para esto se nos presentaron dos posibles soluciones, siendo estas el calcular todas las distancias de las centrales al nuevo punto, y elegir la mínima, y el utilizar la técnica de diagramas de Voroni, o bien de mapas trapezoidales, para localizar el punto más cercano. Para la primera solución, la cual decidimos utilizar, se calculó la distancia de todas las centrales a la nueva contratación, guardando el valor de la mínima. Dado a que no se requería del valor de la distancia, y solo de cual era la mínima, se utilizaron los valores de las distancias cuadradas $d = (dx^2 + dy^2)$. La complejidad de este proceso termina siendo de O(N). La posible mejora sería utilizar algoritmo de Steven Fortune en este problema para realizar mapas trapezoidales, en donde al agregar un punto en el mapa se sabe que este estará mas cercano a la central que le corresponde el trapezoide en el cual se agregó dicho punto. Este algoritmo cuenta con una complejidad de tiempo de O(N log N) y con una complejidad de espacio O(N).

Coevaluación

1. ¿Tu compañero mostró proactividad, colaboración y/o apoyo? (explica por qué)

   Si. Desde un inicio comenzamos por juntarnos en tiempos acordados y en ellos sí estuvo trabajando en el código y planificando sobre qué deberíamos de hacer, así como sobre qué algoritmos deberíamos de usar para diferentes puntos de la actividad.

2. ¿Qué funcionalidades realizó tu compañero? ¿Cuáles hiciste tú? ¿O en todo momento estuvieron trabajando juntos al mismo tiempo?

   En un inicio comenzamos trabajando al mismo tiempo, así como para realizar la lectura del archivo y esqueleto del archivo .cpp, y para determinar qué algoritmos usaríamos, pero una vez con eso determinado el trabajo lo dividimos en cada quien hacer dos de los puntos, por lo cual mi compañero realizó las soluciones a los puntos 1 y 2, y yo realicé las funciones de los puntos 3 y 4.

3. Del 1 al 10, ¿cómo evaluarías a tu compañero?

   10/10.