



Tecnológico de Monterrey

**Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey**

Análisis y diseño de algoritmos avanzados
Grupo 601

Profesor
Luis Humberto González Guerra

Reflexión Individual Evidencia 2

Valeria López Flores A00838648

27 de Noviembre del 2025

Durante este semestre a lo largo de esta clase de Análisis y Diseño de Algoritmos avanzados, hemos trabajado con distintos tipos de algoritmos desde algoritmos de strings hasta algoritmos que utilizan grafos para obtener la distancia mínima entre vértices y muchos más. En esta segunda evidencia integradora utilizamos distintos algoritmos para los cuatro problemas a resolver estos siendo Kruskal, Floyd Warshall, Travelling Salesman Problem con branch and bound y el uso de la distancia euclidiana.

Esta evidencia fue una de las más retadoras que he tenido en cuanto al uso de algoritmos debido a que podría decir que considero como una de mis áreas de oportunidad la implementación de distintos algoritmos con grafos ya que la parte teórica la entiendo muy bien, sin embargo al momento de hacer la implementación programando me toma un poco más tiempo poder hacerlo. Fue un proceso muy interesante ya que tuvimos que pensar un poco fuera de la caja y decidimos utilizar vectores de pares de pares para poder acceder y tener a la mano todos los datos necesarios en el algoritmo de kruskal, para el travelling salesman problem tuvimos que hacer más vectores de los que teníamos en mente inicialmente y para Floyd tuvimos que hacer otra matriz para ir reconstruyendo las rutas y poder hacer el output de manera adecuada y no solo obtener la distancia. Antes de haber cursado esta materia probablemente no hubiera podido pensar en un uso de vectores y de estructuras de datos de esta manera, pero debo de admitir que cada vez me siento más cómoda de experimentar las maneras en las que puedo usar estas estructuras. A continuación va una pequeña explicación de cómo es que adaptamos los algoritmos vistos en clase para dar solución a la evidencia.

Para la primera parte de la evidencia utilizamos el algoritmo de Kruskal para mostrar la manera óptima de cablear, conectando colonias para poder compartir la información con el menor costo tomando en cuenta las conexiones del cableado que ya existen. Iniciamos toda esta evidencia declarando nuestra estructura de Colonias para poder definir los parámetros necesarios de las colonias, estas siendo su nombre, coordenadas en ambas y una declaración de si eran colonias centrales. Asimismo también hicimos un vector de colonias y usamos un hash map (unordered map) para la cuestión del nombre de la colonia con su índice. Esta estructura fue la base para todas las funciones necesarias para la solución de los cuatro problemas.

Utilizamos el código base de Kruskal visto en clase y lo adaptamos a las necesidades del código. Creamos una función para preparar los datos para la función principal de Kruskal en la cual convertimos nuestra matriz de adyacencia en un vector de aristas y así mismo cuando una conexión ya tenía un cableado nuevo le dimos un costo de 0. Finalmente, aplicamos el algoritmo de Kruskal y ordenamos las artistas por costo, e hicimos que se imprimieran las conexiones que necesitan del cableado nuevo.

Después de esto nos enfocamos en el tercer problema de las rutas óptimas entre las centrales. La manera en la que abordamos esto fue basarnos del algoritmo de Floyd Warshall visto en clase y adaptarlo para conocer las rutas completas y no solo el costo. Uno de los principales cambios que hicimos, fue a través de la creación de otra matriz de caminos en la que íbamos guardando los nodos intermedios entre dos nodos. Ahora en el algoritmo base de Floyd

cualquier momento cuando encuentra un camino más corto que el pasado se guarda ese nodo intermedio. Luego para poder reconstruir la ruta completa utiliza la matriz adicional que se hizo previamente. Finalmente en otra función adicional se hace la identificación de aquellas colonias que son centrales y a partir de eso se hace el cálculo de las rutas entre las posibles combinaciones y para imprimir la ruta completa usamos la parte de los nodos intermedios.

Después de haber obtenido correctamente los resultados del algoritmo de Floyd, Comenzamos con la implementación del segundo problema, el Traveling Salesman Problem en el cual fuimos preparando los datos de las colonias no centrales en vectores separados, cada una con un propósito específico como guardar la ruta actual, la más óptima actualmente y otro para ir viendo cuáles colonias han sido visitadas. Luego usamos branch and bound para ir visitando todas las posibles rutas y también se va haciendo la poda de las rutas que no sean mejores que la actual. Si hay una rana que va por un buen camino se marca a la colonia como visitada en el vector específico para esto y vuelve a hacer una llamada recursiva. Luego hay un poco de backtracking para ir explorando las otras ramas. Cuando ya se llevó a cabo esto para las colonias regresa al punto inicial y actualiza lo que se tiene actualmente como la mejor solución. Asimismo algo que fue fundamental para esto fue el uso de Floyd-Warshall para mostrar el camino completo considerando como los nodos intermedios mencionados previamente en la explicación de Floyd para poder dar la ruta específica.

Finalmente para la última parte de la evidencia, el cuarto problema para la conexión de nuevas colonias ya fue mucho más sencillo a todos los demás algoritmos debido a que únicamente fue una implementación de la distancia euclídea entre dos puntos con un enfoque brute force en el que se hace la comparación entre cada colonia nueva con las que ya existen, recordando la distancia mínima. luego lo que hace nuestra función es que imprime la colonia ya existente con la más cercana que se quiere hacer la nueva conexión.