Alejandro Lizárraga Vizcarra
A01284610

Para esta evidencia teníamos que resolver 4 diferentes problemas. Para esto utilizaríamos conceptos que vimos en la clase de algoritmos. Estos temas incluyen, el Minimum expansion tree, Travelling salesman problem, Maximum Flow y Voronoi polygons.

El primer problema del minimum expansion tree, estaba contextualizado en una compañía de redes que quería cablear con fibra óptica una ciudad compuesta de colonias. Para esto leímos un grafo en matriz de adyacencia. Con esto utilizamos el método de Kruskal para resolver el problema. De esta manera pudimos resolver de manera eficiente. Este algoritmo lo pudimos resolver con una complejidad de O(E log(E)) considerando que se necesitaban ordenar un vector al inicio.

Considerando que estamos transaccionando a un mundo digital, aun se necesitan de medios físicos. Para el transporte de estos medios físicos se quiere visitar cada colonia realizando un viaje de menos distancia posible. Esto se le conoce como el Travelling salesman problem. Este problema es conocido por tener una solución de complejidad muy alta por lo que es recomendable resolver con menos de 21 modos si se busca la respuesta optima. La solución implementada es de complejidad O(n!) lo cual como se mencionó previamente, puede resultar un problema si es que se utiliza para muchas colonias o nodos. Este algoritmo al igual que el pasado tomó de entrada una matriz de adyacencia ponderada. Este algoritmo lo resolvimos por fuerza bruta y pudiera ser mejorado con conceptos como Branch and Bound, pero la complejidad aun así resulta muy alta.

Para el problema de Max Flow estamos utilizando el algoritmo de Dinic. Siguiendo con el tema de interconectividad ahora estamos tratando de encontrar el mayor flujo posible de transferencia de datos entre colonias. Esto una vez más fue representado con una matriz de adyacencia con los pesos, pero en este caso el grafo es dirigido, es decir que pueden tener diferentes pesos dependiendo de la dirección que se toma un vértice. La solución implementada tiene una complejidad de O(EV^2), donde V es el número de vértices y E es el número de Aristas. Es importante mencionar que hay otros métodos ayudantes para esta solución sin embargo tienen complejidades menores, por este motivo en un peor caso O(EV^2) sigue siendo la complejidad para la solución de este problema.

Finalmente tenemos el algoritmo de Voronoi. Tomando en cuenta que nos encontramos en un plano dos dimensional contamos con centrales. Estas centrales nos ayudarán a conectar casas a una red. Lo que se intenta resolver en este problema es encontrar polígonos que rodeen las centrales de tal modo que las aristas de los polígonos delimiten un área donde en cualquier punto de esa área uno se encuentre lo más cercano a la central englobada. Este problema lo resolvimos al programar la triangulación de Delunay y utilizando esta información para procesar el método de Voronoi. Considerando que se procesa Delunay junto con Voronoi, al juntar las complejidades de la manera que implementamos el algoritmo, terminamos con una complejidad de O(n^4). Esta complejidad se pudiera buscar mejorar en un futuro y esto pudiera ser con mejor uso de estructuras de datos debido a que varias mentes iteramos sobre vectores para encontrar datos.