

Instituto Tecnológico y de Estudios Superiores de Monterrey
Febrero-Junio 2023



TC 2038
Análisis y diseño de algoritmos avanzados
Grupo 602

Actividad Integradora 2
Reflexión Individual

Nombre y Matrícula:
Samuel Acosta Ugarte | A00833547

Profesor:
Oscar Márquez Calvo, PhD

Fecha:
29 de mayo de 2023

Introducción

En esta actividad integradora, se implementaron 4 algoritmos avanzados que permiten resolver diversos problemas, pero para este caso se implementaron con el propósito de resolver los siguientes problemas:

1. **“Minimum Spanning Tree”**: Desplegar de forma óptima de cablear con fibra óptica conectando colonias.
2. **“Traveling Salesman”**: Conocer la ruta mas corta posible que visita cada colonia exactamente una vez para dejar documentos y regresar a la colonia de origen.
3. **“Maximum Flow”**: Conocer la capacidad máxima de transmisión de datos entre la colonia i y la colonia j, conociendo estimaciones de interferencia.
4. **“Voronoi Diagram”**: Teniendo una cuenta geográfica de varias “centrales” determinar cual es la central mas cercana dada una nueva contratación de servicio.

Minimum Spanning Tree (Algoritmo Kruskal)

Este algoritmo ávaro, dado un grafo ponderado no dirigido encuentra el árbol de mínima expansión. A grandes rasgos empieza con un grafo vacío y agrega la siguiente arista con la mínima ponderación, siempre y cuando el grafo que se esta creando no arme un ciclo. La implementación del algoritmo implementado es de: $O(n \cdot \log n)$

Traveling Salesman

El problema del agente viajero se encarga de encontrar la distancia mas corta que un viajero puede tomar al visitar un conjunto de ciudades, empezando por una ciudad, recorrer todas las ciudades y regresar a donde empezó. En general se comienza en una ciudad arbitraria y visita la ciudad mas cercana no visitada y cambia el estado de la ciudad actual como visitada. Básicamente se repite hasta haber visitado todas las ciudades. La implementación del algoritmo implementado es usando la estrategia de fuerza bruta cuya complejidad es: $O(n!)$ donde n es la cantidad de colonias en este caso.

Maximum Flow

Para este problema se uso el algoritmo de Dinic donde a partir de un grafo no dirigido, encontrar el mayor flujo posible entre las colonias como mencionado anteriormente. Como la implementación recibe una matriz de adyacencia la complejidad es: $O(c \cdot v^2)$ donde a es el numero de aristas y c es la cantidad de colonias.

Voronoi Diagram

El diagrama de Voronoi se basa en otro algoritmo para esta implementación llamada Triangulación de Delaunay. Hablemos primeramente de este algoritmo

La triangulación de Delaunay tiene como propósito regresar, a partir de una lista de puntos, triángulos, cuyo circuncírculo no incluya ninguno de los demás puntos. Por cada punto que se quiera agregar, se calculan los circuncírculos de cada triángulo que exista en el momento. Si se encuentra que el punto a agregar está dentro de uno o más triángulos, estos triángulos son inválidos y se quitan las aristas que comparten cada uno de esos triángulos, deshaciéndolos. Lo que sigue es formar aristas a partir de cada

uno de los puntos de los triángulos inválidos que compartan aristas con el punto que se está agregando. Al final del algoritmo tenemos la triangulación completa. Los circuncentros de estos triángulos nos ayudan a formar el diagrama de Voronoi. La complejidad de nuestra implementación de la triangulación de Delaunay es **$O(n^4)$** donde n es la cantidad de puntos que se quieren procesar para formar la triangulación completa.

En cuanto a Voronoi, se usa el algoritmo de triangulación de Delaunay para sacar los circuncentros de cada uno de los triángulos en la triangulación. El propósito de esto es poder interconectarlas siempre y cuando los triángulos compartan aristas. La interconexión de estos circuncentros genera el diagrama de Voronoi. La complejidad de nuestra implementación de Voronoi es **$O(n^4)$** debido al costo alto de Delaunay.

Conclusión

Esta actividad integradora 2 fue realmente retadora para mi y mi equipo. En especial, la implementación del diagrama de Voronoi. Sin embargo, fue una experiencia enriquecedora y mejoro mucho la manera en que implemento y diseño algoritmos para la solución de problemas reales.