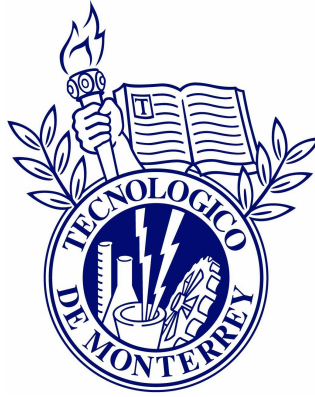


INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY



Escuela de Ingeniería

Análisis y diseño de algoritmos avanzados (Gpo 2)

Reflexión individual de la “Actividad Integradora 2”

Presenta:

Carlos Noel Ojeda Angulo, Matrícula A01741085

Guadalajara, Jal., 12 de Noviembre de 2021

Reflexión individual de la “Actividad Integradora 2”

En el presente documento se encuentra mi reflexión acerca de la solución a la que mi compañera Ana Cristina Munguía Romero y yo (Carlos Noel Ojeda Angulo) llegamos para resolver la situación problema 2.

Al analizar la problemática planteada comprendimos que necesitamos leer lo que contiene un archivo (llamado *input.txt*) y obtener un punto (x, y) desde la consola, para procesar su contenido y dar solución a las cuatro secciones de la que consta la *Actividad Integradora 2*.

La primera parte de la actividad consiste en encontrar la forma óptima de cablear la ciudad, para que todas las colonias puedan comunicarse. Esto se hace a partir de la lectura de un archivo que contiene entre otras cosas, una matriz de adyacencia, donde se representa la distancia en kilómetros entre dichas colonias. Posteriormente se procesa esta información, y se encuentra la forma óptima de cablear la ciudad al imprimir la menor distancia entre cualesquiera dos colonias, con el uso del algoritmo de *Floyd-Warshall*, el cual tiene una complejidad temporal $O(n^3)$, debido a que no solamente buscamos el camino más corto entre un par de colonias (como en el algoritmo de Dijkstra) buscamos el camino más corto entre cada posible par de colonias. Decidimos implementar este algoritmo (y no el algoritmo de *Dijkstra*) porque ahorra espacio en memoria al sobreescribir lo que se procesa en la misma matriz. Es importante mencionar que para imprimir las colonias con caracteres alfanuméricos añadimos un arreglo, el cual al aplicar módulo y una pequeña condicional nos permite dar este formato a las colonias.

En la segunda sección de la actividad también hacemos uso de una matriz de adyacencia que representa la distancia en kilómetros entre las colonias, a partir de ella encontramos una solución pseudo-óptima para recorrer todas las colonias. Implementamos el algoritmo *Repetitive Nearest-neighbor*, el cual encuentra una solución con una complejidad temporal razonable $O(n^3)$, como se mencionó anteriormente, no siempre se encuentra la solución óptima porque implementamos un algoritmo de tipo avaro (encuentra mínimos locales, y en ocasiones globales). Este algoritmo tiene una ventaja en comparación al *Nearest Neighbor*; porque selecciona el camino más corto que se encontró al aplicar todas las colonias como iniciales en *Nearest Neighbor*.

La tercera parte de la actividad consiste en encontrar el máximo flujo de información que puede haber entre dos colonias, dada una matriz de adyacencia (en *input.txt*) que representa la capacidad máxima de transmisión entre cualesquiera dos colonias. En esta actividad seleccionamos la colonia A como inicial y N-1 como final. Para ello implementamos como base el algoritmo de *Ford-Fulkerson*, y lo modificamos un poco al añadir *Breath First Search* para determinar si aún es posible aumentar el flujo, este algoritmo es llamado *Edmonds-Karp* y tiene una complejidad temporal $O(V^3 \cdot E)$, donde V es el número de vértices y E de arcos.

Para la última parte, necesitábamos encontrar la central más cercana dada (por medio de la consola) la ubicación de un establecimiento o casa que quiere contratar servicios de internet. La ubicación de las centrales se obtiene de *input.txt*, decidimos implementar un diagrama de Voronoi con CGAL, esta estructura nos permite determinar dado un nuevo punto, en qué cara o región se encuentra e imprimir los vértices que conforman esta cara o región, es decir cuál es la central más cercana. Debido a que no encontramos la complejidad temporal de esta implementación en la documentación de CGAL, y dada una charla con nuestro docente, creemos firmemente que la complejidad temporal para la construcción de este algoritmo es igual o inferior a $O(n * \log(n))$, donde n es el número de centrales.

Para finalizar la reflexión, puedo decir que lo primordial en la actividad fue considerar diversas variables y decidir qué algoritmos implementar. Ya que esto nos permitió desarrollar algoritmos eficientes, en el menor tiempo posible y disfrutar del proceso al hacerlo en *pair programming* (programación en pareja).