

Reflexión Actividad Integradora 2



**Tecnológico
de Monterrey**

Fecha de la Entrega: 29/11/2023

Alumnos

Fernanda Osorio A01026502

Ian Holender A01026295

Lucia Barrenechea - A0178202

En el caso de la problemática 2, abordamos una serie de desafíos que requerían la aplicación de diversas técnicas y algoritmos, entre ellos el método de branch and bound, el algoritmo de Prim, Fuerza bruta, entre otros. La colaboración en equipo fue fundamental para agilizar el proyecto, aprovechando nuestros conocimientos diversos que se complementan mutuamente.

La primera solución propuesta para esta actividad fue utilizar el Algoritmo de Prim para desplegar la forma óptima de cablear con fibra óptica las colonias con el fin de conectarlas todas.

Se utilizó el algoritmo de Prim ya que este genera un MST (minimum spanning tree). Este algoritmo todavía entra dentro de la categoría “greedy” sin embargo este si encuentra la solución correcta. El algoritmo de Prim entra dentro la categoría greedy ya que en todo momento hace decisiones que lo pueden llevar a encontrar la solución más óptima (pequeña). En otras palabras, la estrategia de este algoritmo de siempre seleccionar el nodo más pequeño lleva a la solución más óptima del problema.

Así fue como utilizando este algoritmo se puede encontrar la mejor manera (más corta) de cablear (conectar) todas las colonias.

Esta implementación utiliza 3 funciones principales: *minKey*, *printMST* y *Prim*. La función *minKey* tiene una complejidad $O(V)$, donde V son los vértices, ya que esta función itera sobre todos los vértices una vez para regresar el índice del vértice con la distancia mínima.

La función *printMST*, se encarga de imprimir el MST. También tiene una complejidad $O(V)$ ya que itera sobre todos los vértices una vez. La función *Prim* tiene una complejidad $O(V^2)$ ya que itera sobre V una vez, dentro esa iteración llama a *minKey* ($O(V)$) por lo que su complejidad es de $O(V^2)$.

Entonces en general la complejidad de esta implementación es $O(V^2)$. Resolver esta parte de la actividad integradora nos permitió ver la importancia de un MST y como este se puede utilizar. También fue una gran ayuda en poder reconocer la diferencia entre un algoritmo totalmente greedy y un algoritmo como el de Prim, que a pesar de entrar en esa categoría si busca obtener la mejor solución.

La segunda parte del problema involucra abordar el "Travelling Salesman Problem", que consiste en visitar cada ciudad exactamente una vez, de la manera más eficiente posible, y al finalizar, regresar a la ciudad inicial. Aunque nos llevó tiempo plasmar nuestras ideas en código, estamos muy satisfechos con el resultado obtenido. Utilizamos la técnica branch and bound para analizar las posibles soluciones. Para resolver este problema utilizamos 4 funciones diferentes. Estas funciones tienen complejidad de $O(n)$ y complejidad de $O(n^2)$ esto significa que la complejidad del header tsp tiene una complejidad algorítmica de Big O de $O(n^2)$. Entre más grande sea el grafo, más tiempo tardara en ejecutarse.

La tercera parte resuelve el problema de encontrar la central más cercana a un nuevo servicio, dados los datos de las centrales y el nuevo servicio. Se emplea la técnica de fuerza bruta para calcular la distancia euclidiana entre el nuevo servicio y cada central, y luego se identifica la central más

cercana. La función *calculateDistance* calcula la distancia euclidiana entre dos puntos, mientras que *findNearestCentral* encuentra la central más cercana utilizando un enfoque de comparación iterativa. La complejidad del código se estima en $O(M^2 + N)$ donde "M" es el tamaño de la matriz y "N" es el número de centrales. Esta complejidad proviene de la lectura de una matriz cuadrada de tamaño "M" y del procesamiento de "N" centrales mediante el algoritmo de fuerza bruta.

Reflexiones individuales:

Fer:

El curso en su totalidad y ambas actividades integradoras me permitieron entender porqué es importante conocer en qué momentos aplicar qué algoritmos. Muchas veces el algoritmo más optimizado parecerá la mejor opción pero siempre hay que tener en cuenta como se acopla al problema. A veces un algoritmo más rudimentario puede generar una complejidad Big O menor que un algoritmo avanzado. Es por esto que la resolución de las actividades integradoras fue una gran herramienta de aprendizaje.

Los contextos de ambas actividades demuestran la importancia del diseño y aplicación de algoritmos avanzados en la vida cotidiana. Por ejemplo en la segunda actividad menciona como la pandemia generó una demanda más grande por servicios de instalación de internet, para poder completar esa tarea de manera correcta claramente es necesario conocer las mejores maneras de conectar todas las colonias.

De este curso me llevo siempre considerar las características del problema para considerar su mejor solución y siempre tomar en cuenta las normas para llevar a cabo la implementación de la mejor manera.

Lu:

Durante el semestre, exploré diversos algoritmos y sus aplicaciones. La diversidad de enfoques para abordar problemas similares resaltó la capacidad constante de generar soluciones innovadoras. La importancia de evaluar todas las opciones disponibles se reveló como un paso crucial para garantizar que no se pase por alto una implementación más efectiva. Esta experiencia me brindó la habilidad de aplicar mis conocimientos en situaciones cotidianas, profundizando mi comprensión sobre la necesidad de crear algoritmos innovadores.

Ian:

Después de haber concluido el curso, obtuve conocimientos sobre diversos tipos de algoritmos, incluyendo aquellos relacionados con búsqueda, grafos, recorridos, cadenas de caracteres, entre otros. Aprendí cómo aplicarlos de diversas maneras para abordar problemas específicos. Además, durante el

curso, desarrollé un razonamiento sólido para la implementación de los mismos, ya que comprendí que, en algunos casos, el algoritmo más eficiente no necesariamente es la solución óptima para un problema dado. También logré entender la importancia de la notación Big(O), ya que esta complejidad nos permite evaluar cuántos recursos consumirá nuestro programa, brindándonos una perspectiva crucial para tomar decisiones en relación con los algoritmos utilizados.

El curso logró cambiar por completo mi perspectiva sobre el razonamiento lógico en el desarrollo de mis futuros códigos. Me preparó para cuestionar siempre cuál es la opción más eficiente para abordar los problemas que pueda enfrentar en el futuro.