TECNOLÓGICO DE MONTERREY
Campus Querétaro

PROFESOR ENCARGADO:

Ramona Fuentes Valdéz

Análisis y diseño de algoritmos avanzados (Gpo 602)

Situación Problema 2

Presenta
Daniel Gutiérrez Gómez A01068056
Juan Manuel González Ascencio A00572003
Julio César Pérez Rodríguez A01705763

FECHA DE ENTREGA:
18/11/2023

Durante el segundo periodo y lo que va del tercer periodo del semestre, hemos estado viendo temas relacionados con la teoría de grafos, los cuales como sabemos tienen demasiadas aplicaciones en la vida cotidiana. Los grafos son estructuras discretas que constan de vértices y aristas que conectan entre sí esos vértices.
Un grafo denominado *G* consta de dos partes:

1. Un conjunto *V = V(G)* cuyos elementos se denominan vértices, puntos o nodos de G.
2. Un conjunto *E = E(G)* de pares de vértices distintos denominados aristas de *G.*

De manera muy amplia, se puede decir que existen dos tipos de grafos, los grafos dirigidos y los grafos no dirigidos. Los grafos dirigidos son aquellos en los que de una manera menos abstracta, en su representación física por así decirlo, son aquellos en los que sus aristas llevan una dirección de hacía que vértice se quiere llegar desde un nodo orígen. De esta manera se pueden observar las conexiones de todos los vértices en conexión con otros sabiendo el punto de partida y punto de llegada.

Por lo tanto, aquellos grafos que no llevan dirección se toma por hecho que puede llegar a existir conexión entre dos vértices sin tomar en cuenta el sentido que llevan y sin saber un orígen exacto ni un destino exacto, simplemente se sabrá que dos vértices tendrán conexión pero sin esa especificación como tal.

Si ahondamos más en la teoría de grafos, veremos que existen más tipos de grafos, estos son:

- Grafos de cadena
  Tipo de grafo híbrido formado por aristas dirigidas y no dirigidas.
- Grafos simples
  Tipo de grafo el cual no incluye ciclos ni aristas paralelas.
- Multigrafo
  Son grafos con dos o más aristas que pueden conectar a un mismo vértice.
- Grafo completo
  Tipo de grafo con aristas entre cada par de vértice
- Grafo bipartito
  Tipo de grafo que se pueden dividir en dos subconjuntos disjuntos de vértices, donde cada una de las aristas conecta a un vértice del primer conjunto con uno del segundo.
- Grafo pesado
  Tipo de grafo que tiene pesos asociados a vértices y/o aristas.

Después de una pequeña introducción de lo que son los grafos, trataremos de dar solución por medio de algoritmos basados en la teoría de grafos a la siguiente problemática:

Durante el año 2020 todo el mundo se vio afectado por un evento que nadie esperaba: la pandemia ocasionada por el COVID-19. En todos los países del planeta se tomaron medidas sanitarias para intentar contener la pandemia. Una de estas medidas fue el mandar a toda la población a sus casas, moviendo gran parte de las actividades presenciales a un modelo remoto en el que las empresas proveedoras de servicios de Internet (ISP por sus siglas en inglés de Internet Service Provider) tomaron un papel más que protagónico. Mucha gente se movió a la modalidad de trabajo remoto, o home-office, también la mayoría de instituciones educativas optaron por continuar sus operaciones bajo un modelo a distancia aumentando de gran forma la transmisión de datos en Internet.

Si estuviera en nuestras manos mejorar los servicios de Internet en una población pequeña,

¿Podríamos decidir cómo cablear los puntos más importantes de dicha población de tal forma que se utilice la menor cantidad de fibra óptica?

Asumiendo que tenemos varias formas de conectar dos nodos en la población,

Para una persona que tiene que ir a visitar todos los puntos de la red, ¿Cuál será la forma óptima de visitar todos los puntos de la red y regresar al punto de origen?

¿Podríamos analizar la cantidad máxima de información que puede pasar desde un nodo a otro ?

¿Podríamos analizar la factibilidad de conectar a la red un nuevo punto (una nueva localidad) en el mapa ?

Como hemos visto la problemática y la hemos analizado, se nos ha planteado una serie de problemas que van estrechamente relacionados con las preguntas que se nos aborda en la situación problema. Se busca solucionar estas problemáticas con el tema visto en la introducción que son la implementación de grafos y los diferentes algoritmos que pueden existir para determinada problemática dentro de la teoría de grafos.

Se nos solicita que leamos un archivo de entrada con información importante acerca de las colonias que están dentro de una ciudad y que dichas colonias están conectadas entre sí, por medio de distancias en unidades de kilómetros, el proveedor de servicios busca una forma ***óptima*** de establecer un cableado con fibra óptica a lo largo de la ciudad con el objetivo de conseguir una conexión entre colonias colindantes con la menor cantidad de cable posible. Para esta primera problemática hemos decidido como equipo el realizar un algoritmo que es bastante eficiente en la búsqueda de un grafo ponderado, en el que se busca recorrer todos los vértices y se busca la ***optimización,*** dicho algoritmo es el de Prim 's. Con este algoritmo conseguiremos establecer una ruta la cual el empleado del proveedor de servicios deberá seguir para poder obtener una ***optimización*** en el uso del cable

Nuestro resultado fue el esperado y además hemos conseguido tener un algoritmo bastante eficiente, en la cual tenemos una complejidad de tiempo de **O(E*log(E))**, donde ***E*** simboliza el número de aristas. El ***máximo*** número de aristas puede ser calculado por medio de una fórmula matemática, la cual únicamente sirve en aquellos grafos en los que se tiene una conexión total de todos los vértices. Dicha fórmula dice lo siguiente: $E = \frac{V(V-1)}{2}$ donde V indica cantidad de vértices. Esta fórmula propone que cada arista conecta dos vértices, y que además cada vértice puede ser conectado a $V - 1$ vértices debido a que se excluye el vértice actual. En este caso se está tomando el cálculo de las aristas dos veces, por lo cual se divide entre dos.

Ahora, para la segunda problemática se nos plantea que, debido a que las ciudades apenas están incorporándose a un mundo tecnológico, se requiere que un trabajador del proveedor de servicios de internet, visite cada colonia para ir dejando estados de cuenta en físico, publicidad, avisos y notificaciones impresas. Para esto, se quiere recorrer este camino desde una colonia de origen recorriendo todas las colonias y regresar al punto de partida con la ruta más corta. Hemos decidido darle solución como equipo con el uso de un algoritmo muy sonado en el mundo de la ingeniería en sistemas y área de computación, el cual es un algoritmo de tipo NP Hard, que significa que no se ha encontrado una solución polinomial al problema. Esto también nos habla de lo complicado que puede llegar a ser el resolver un problema con el uso de este algoritmo. Este algoritmo de TSP(Traveling Salesman Problem), puede resolverse con varios métodos, métodos como Branch and Bound, reducción de matriz,programación dinámica, existen más métodos, sin embargo nosotros nos enfocamos y decidimos utilizar el método de programación dinámica. Este algoritmo con el uso de la programación dinámica tiene una complejidad de tiempo de **O(n!).** Básicamente este tiempo es de los peores que puede existir, sin embargo, nos da una solución precisa. Con este programa hemos obtenido los resultados esperados, recorriendo todas las colonias y regresando al origen por medio de la ruta más cercana.

La empresa quiere conocer el flujo máximo de información del nodo inicial al nodo final. Esto debe desplegarse también en la salida estándar.

Para poder resolver este problema utilizamos el algoritmo de Ford Fulkerson, el cual nos permite calcular el flujo máximo de un nodo inicial al nodo final. El algoritmo funciona iterando sobre un camino de aumento, en el cual, cada camino tiene un flujo máximo y nos deja un grafo residual. El grafo se obtiene restando el flujo actual de la capacidad que tiene cada camino, el algoritmo hace que incremente este flujo a lo máximo posible, la cual es la capacidad mínima de los bordes a lo largo del camino.

La empresa quiere contar con una forma de decidir, dada una nueva contratación del servicio, cuál es la central más cercana geográficamente a esa nueva contratación.

Para este último problema, hemos decidido que, para saber en donde se agregaría esta nueva central, dadas las ubicaciones de las ya establecidas y que se quieren poner lo más cerca posible, que lo más óptimo sería realizar el cálculo de la ruta más cercana por medio de las matemáticas y la fórmula de la distancia euclídea, que básicamente es la raíz cuadrada de la suma al cuadrado de cada punto de los pares coordenados dados. En este caso, al realizar esta operación de comparación para cada posición de los servicios ya establecidos con la nueva contratación se obtiene una complejidad de **O(n).** Esto porque como tenemos un contenedor con las posiciones actuales, y con la nueva posición, para cada posición del contenedor se hace la comparación con la nueva posición y al final se obtiene únicamente la que nos devuelva la menor cantidad de distancia entre ambos puntos de comparación en ese instante de tiempo.

Ahora, retomando las preguntas planteadas en un inicio de la situación problema dos y abordando los algoritmos utilizados, daremos respuesta a las preguntas :
Si estuviera en nuestras manos mejorar los servicios de Internet en una población pequeña,

¿Podríamos decidir cómo cablear los puntos más importantes de dicha población de tal forma que se utilice la menor cantidad de fibra óptica?
Sí, con el algoritmo de Prims, se pueden calcular los puntos en los cuales se debería de iniciar por cablear, así como por donde terminar, para utilizar la menor cantidad de fibra óptica y que así el costo del cable de fibra óptica sea menor, con este algoritmo eficiente en tiempo de complejidad O(Elog(E)).

Asumiendo que tenemos varias formas de conectar dos nodos en la población,

Para una persona que tiene que ir a visitar todos los puntos de la red, ¿Cuál será la forma óptima de visitar todos los puntos de la red y regresar al punto de origen?
Para esto, la solución más óptima para saberlo en un espacio representado por nodos, lo que se puede utilizar de algoritmo es el de TSP (travelling salesman problem), lo cual es un algoritmo que te permite visitar todos los nodos de un espacio y regresar al punto origen con la menor cantidad de "pasos", en este caso menor cantidad de distancias.

¿Podríamos analizar la cantidad máxima de información que puede pasar desde un nodo a otro?
Sí, sin lugar a dudas con el uso del algoritmo de Ford Fulkerson, el cual nos ayuda a obtener el flujo de información desde un punto inicial a uno final, y obtener así la cantidad total de información que pasa a través de un espacio representado por nodos conectados.

¿Podríamos analizar la factibilidad de conectar a la red un nuevo punto (una nueva localidad) en el mapa ?
Sí, con el uso de algoritmos computacionales basados en la geometría, en nuestro caso decidimos utilizar una fórmula matemática (distancia Euclídea) y aplicarla para cada punto junto con la coordenada de la nueva central que se quiera conectar, con esto, al final obtenemos una distancia mínima, así como en que central se deberían de posicionar para conectar esa nueva central.

Conclusiones

Daniel Gutiérrez Gómez

Como conclusión, puedo decir que la teoría de grafos es de suma importancia para nosotros los ingenieros de software, pues se pueden aplicar en la vida diaria en los retos que se nos van presentando, además de esto, es interesante que, quien sabe, igual en algún punto alguien descubre un algoritmo que es más óptimo que los que ahorita ya existen (cualquier tipo de problema del tipo NP-Hard) y que, con esto, se explote aún más el amplio uso y el impacto positivo que está obtención tendría en las personas. Como lo dije, no solo en esta situación problema se aplica la teoría de grafos sino que también, en nuestro día a día, los celulares que utilizamos utilizan algoritmos de ordenamiento cuando se necesita, aplicaciones como google maps, utilizan el algoritmo de TSP, con servidores estupendos que permiten realizar estas operaciones en segundos, para cantidades de datos enorme, y que de no ser un algoritmo eficiente el que es utilizado para este tipo de aplicaciones, los clientes no usamos esa aplicación.

Juan Manuel González Ascencio

Este proyecto fue un reto para mí, debido a que tuve que lograr comprender los algoritmos en profundidad para poder implementarlos de manera exitosa, tuve que primero desarrollar las respuestas en papel y ya después pasarlas a código. Uno de mis aprendizajes más grandes, después de la parte técnica y de ver las soluciones que proponían estos algoritmos, fue el poder comprender donde es que se pueden aplicar en la vida real, con este reto pude relacionar de manera exitosa como es que estos algoritmos pueden tener un impacto en empresas de la vida real. Despues de hacer este proyecto puedo afirmar que tengo un mejor dominio sobre los algoritmos y la resolución de problemas en general y soy capaz de identificar situaciones y escenarios en la vida real donde se puedan aplicar.

Julio Cesar Perez Rodríguez

Este proyecto me pareció una buena manera de poner en práctica los algoritmos en un problema real, para mí este trabajo no solamente representó una actividad académica sino también un reto para poder buscar la mejor forma de implementar una solución, aunque en este trabajo no implementamos las mejores soluciones que encontramos, en la última problemática utilizamos los diagramas de voronoi pero esto lo utilizamos debido a que era un algoritmo libre, para los otros utilizamos algunos algoritmos específicos que habíamos visto en clase aunque también encontramos que habría algo unos algoritmos que serían más eficientes para resolver los problemas debido a que algunos algoritmos tenían una complejidad que en caso de que las colonias aumentaran se aumentaría su tiempo de una manera exponencial.

Esta actividad también me ayudó a ver ejemplos claros en los cuales los algoritmos que aprendemos se pueden utilizar en la vida real ya que muchas veces al verlo solo en teoría pues parece algo que no tendría una aplicación en la vida real pero me doy cuenta que en la vida cotidiana hay muchos sistemas y muchas cosas que se repiten día tras día y que se pueden implementar de una manera algorítmica y así solucionar y reducir el tiempo para realizar ciertas actividades de la vida cotidiana.

Referencias bibliográficas

Ciencias Computacionales. (s. f.). posgrados.inaoep. Recuperado 12 de noviembre de 2023, de https://posgrados.inaoep.mx/archivos/PosCsComputacionales/Curso_Propedeutico/Matematicas_Discretas/Capitulo_4_Grafos.pdf

Abdul Bari. (2018). 3.5 Prims and Kruskals Algorithms - Greedy Method [Vídeo]. YouTube. Recuperado 12 de noviembre de 2023, de https://www.youtube.com/watch?v=4ZlRH0eK-qQ

GeeksforGeeks. (2023b, agosto 4). PRIM S Algorithm for Minimum Spanning Tree MST. https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/

The Editors of Encyclopaedia Britannica. (2023, 6 octubre). NP-Complete Problem | Definition, Examples, & Facts. Encyclopedia Britannica. https://www.britannica.com/science/NP-complete-problem#:~:text=NP%2Dcomplete%20problem%2C%20any%20of,%2C%20and%20graph%2Dcovering%20problems.