Análisis y diseño de algoritmos avanzados (Gpo. 601)

E2. Actividad Integradora 2 - Reflexión

20 de Octubre, 2022
ITESM Puebla

Jorge Angel Delgado Morales (A01551955)

Tecnológico de Monterrey

A lo largo de este periodo hemos aprendido distintos algoritmos para resolver problemas relacionados con grafos, evidentemente cada uno sirve un fin distinto a los demás. Para poder resolver los problemas planteados en esta actividad, era necesario identificar cual era el algoritmo que nos vendría de mayor utilidad.

Para el primer problema, se nos pedía hallar una "ruta" o forma de cablear todas las colonias, de manera que todas estén conectadas (no necesariamente de forma directa). Básicamente lo que queríamos hacer aquí era encontrar el "minimum spanning tree", esto significa de forma muy simplificada, que dado nuestro grafo, queremos quitar todas las aristas que podamos, y que al terminar el grafo siga conectando todos a todos los nodos y que la suma de las aristas que se conservaron sea la menor posible, ya que el peso de estas aristas representa la cantidad de cableado necesario para conectar 2 colonias. Para hacer esto usamos el algoritmo de Prim, el cual hace exactamente esto de una forma "greedy". Este tiene una complejidad temporal de O(V ** 2), donde V es el número de vértices en el grafo, o en nuestro caso las colonias que se busca conectar.

Para el segundo problema, teníamos que encontrar una ruta que terminase en el inicio, la cual debía de cumplir con 2 condiciones; solo pasar por una colonia una vez y que se visitaran todas las colonias. Para esto hicimos algo muy parecido al problema anterior, aplicamos Prim para todos los nodos, y luego checamos cual es el menor costo para ir a otro nodo, si es a partir de uno que no es el que tenemos como actual, a ese costo le sumamos el costo de llegar a ese otro costo. Para esto tenemos una función que se encarga de encontrar cuál será el costo entre 2 nodos dados, y otra función que calcula cual sera el siguiente nodo al que hay que moverse (para esta vemos de las adyacencias cual tiene menor costo). Ambas funciones tienen una complejidad temporal de O(n) donde n es el número de nodos. Estas son llamadas desde otra función auxiliar, cada una dentro de un ciclo de complejidad O(n). Por lo que esta función auxiliar (que es la que se llama desde main) tiene una complejidad O(n ** 2).

En el tercer problema tenemos que encontrar el flujo máximo. Para esto realizamos un BFS, el cual nos ayuda a determinar si una ruta es válida y si lo es poder sumar los valores que representan ese recorrido. Obtener la complejidad fue algo costoso, ya que se realizaron varias operaciones de distintas complejidades, pero asumimos que es O(N ** 5) donde n es el numero de nodos, ya que el BFS es O(n ** 2) y se usa como condición en un ciclo O(n), el cual ejecuta operaciones de tipo O(n ** 2).

Finalmente, teníamos que encontrar a que central conectar un nuevo nodo dadas sus distancias hacia estas. Para esto calculamos la distancia entre estas y el nodo y usamos la de menor distancia. Esto tiene una complejidad O(n) donde n es el numero de nodos o centrales. Ya que queremos checar la distancia entre el nodo dado y todas estas y agarrar el de menor distancia.