Instituto Tecnológico y de Estudios Superiores de Monterrey.
Campus Monterrey

Escuela de Ingeniería y Ciencias

Análisis y diseño de algoritmos avanzados (Gpo 570). TC2038.570

E2. Actividad Integradora 2
Reflexión

Dirigido a:
Felipe Castillo Rendón

Por:
Oscar Arreola Alvarado | A01178076

1 de Diciembre del 2024
Monterrey, Nuevo León.

Reflexión

En este proyecto tuve la oportunidad de explorar diversos algoritmos interesantes que sirven para resolver problemas de la vida real. Fue interesante investigar acerca de las diversas formas de resolver los problemas y ver cómo es que algunos algoritmos hacen optimizaciones de manera inteligente para resolver los problemas con mayor eficiencia.

Para resolver eficientemente la evidencia, fue necesario llevar a cabo la implementación de 4 diferentes algoritmos, cada uno resolviendo una de las problemáticas especificadas por la evidencia.

Para el primer problema, que consistía en encontrar la forma óptima para conectar todas las ciudades, se implementó el algoritmo de Kruskal. Dicho algoritmo permite, de manera eficiente, encontrar el Minimum Spanning Tree (MST). El problema consiste en hallar la forma de menor costo en que se pueden conectar todos los nodos de un grafo. El algoritmo de Kruskal tiene una complejidad temporal de O(E * logE): la idea del algoritmo es seguir un enfoque greedy donde primero se ordenan todos los edges y se elige el de menor costo, en caso de que este resulte en un nuevo nodo unido al árbol actual.

En segundo lugar, se utilizó el algoritmo clásico del traveling salesman. Este algoritmo permite encontrar la forma más barata de recorrer una serie de nodos y regresar al nodo inicial. Para la implementación, se hizo uso de la programación dinámica para usar los resultados computados previamente para futuros problemas, al igual que una matriz para guardar el camino recorrido y reconstruirlo al final del algoritmo. El algoritmo final tuvo una complejidad temporal de O(n^2 * 2^n), haciendo uso de la memorización. La eficiencia del algoritmo es bastante superior al enfoque “naive” que consiste en encontrar todas las permutaciones y guardar el resultado más barato, que tiene una complejidad de O(n!). Implementar este algoritmo fue bastante interesante pues demuestra cómo puede mejorar enormemente la solución dependiendo del enfoque seleccionado.

En tercer lugar, se utilizó el algoritmo de Ford Fulkerson para encontrar la capacidad máxima de transmisión entre dos colonias dada una matriz de adyacencias. El algoritmo consiste en buscar caminos desde el nodo inicial al final que tengan “flow” positivo. En caso de tenerlo, se añade el flow a la respuesta y se computa el grafo residual, que surge al restar el flow en el orden inverso del camino. El algoritmo resulta bastante interesante y parece tener una variedad de aplicaciones en la vida real. Cabe a destacar que el algoritmo tiene una complejidad temporal de O(max_flow * e), donde max_flow es la cantidad máxima de flujo que se puede pasar y e la cantidad de edges en el grafo.

Finalmente, para resolver el último problema, se utilizó una búsqueda lineal, con complejidad temporal O(n). Aunque la solución no fue tan complicada de implementar, también resultó interesante ver cómo en ocasiones un algoritmo simple puede ayudar a resolver un problema eficientemente.

En general, la evidencia resultó en una variedad de aprendizajes y nuevamente demuestra lo mucho que vale la pena la búsqueda de algoritmos eficientes para resolver problemas en la vida cotidiana.