Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Análisis y diseño de algoritmos avanzados

Reflexión E2- Actividad Integradora 2

Rodolfo Emiliano Salazar Rodarte A01198201

Aldo Leonardo Lopez Ruiz A01254678

Felipe de Jesus Ramos Santos A00837590

Repositorio: Foferr/ACT_INT2

Foferr/ACT_INT2

A lo largo de este proyecto, trabajar con algoritmos como Kruskal, BFS (Breadth-First Search), TSP (Travelling Salesman Problem) y Linear Search ha representado un verdadero desafío y una oportunidad de crecimiento. Cada uno de estos algoritmos aporta una perspectiva distinta sobre cómo resolver problemas complejos de manera eficiente, y su implementación nos permite reflexionar sobre su aplicabilidad en situaciones reales.

Kruskal es un algoritmo fundamental en la teoría de grafos para encontrar árboles de expansión mínima. Su implementación me permitió entender la importancia de la estructura de datos y cómo las uniones y búsquedas en conjuntos disjuntos son clave para optimizar el rendimiento. La experiencia de codificar este algoritmo me mostró cómo pequeños detalles en la selección de las aristas pueden marcar la diferencia en la eficiencia general.

Por otro lado, BFS fue una herramienta invaluable para explorar grafos de manera sistemática, asegurando que se recorrieran todos los nodos de manera ordenada. Este algoritmo me llevó a profundizar en la gestión de colas y a pensar en soluciones a problemas de búsqueda, destacando su aplicación práctica en escenarios como la búsqueda de rutas óptimas o la exploración de redes.

El Travelling Salesman Problem (TSP), aunque más teórico y con una complejidad computacional considerable, me enseñó la importancia de los enfoques heurísticos y aproximados en problemas donde encontrar la solución exacta no es siempre viable. Al enfrentar este algoritmo, entendí cómo las decisiones locales pueden influir en el resultado global y cómo balancear precisión y tiempo de ejecución.

Finalmente, el Linear Search, aunque básico en comparación con los otros algoritmos, reafirmó la importancia de entender la complejidad temporal de las operaciones más simples. Este algoritmo fue un recordatorio de que incluso las

tareas más triviales pueden convertirse en cuellos de botella si no se manejan adecuadamente. A01254678

En este proyecto, hemos implementado varios algoritmos clásicos, y aunque cada uno tiene su propio propósito, todos tienen algo en común: buscan resolver problemas complejos de manera eficiente. Por ejemplo, el algoritmo de Kruska**l** nos ayuda a encontrar el árbol de expansión mínima de un grafo, lo cual es esencial para cosas como la optimización de redes. Al hacerlo, usamos un enfoque basado en conjuntos disjuntos para mantener el seguimiento de las conexiones y evitar ciclos, lo que hace que el algoritmo sea bastante elegante y eficiente para grafos grandes.

El algoritmo de flujo máximo es otro que siempre me llama la atención. Resolver problemas de flujo en redes, como determinar la cantidad máxima de material que puede pasar de un punto a otro en una red de transporte, tiene un montón de aplicaciones prácticas, como en la optimización de redes de suministro. Aquí, aplicamos el enfoque de Ford-Fulkerson, que es bastante directo pero, a veces, se vuelve algo costoso en términos de tiempo si no usamos mejoras como el BFS

El algoritmo de búsqueda lineal es el más sencillo, pero incluso con algo tan básico como esto, siempre es bueno recordar cómo las soluciones simples pueden ser muy efectivas en ciertos casos. En proyectos más grandes, puedes recurrir a algoritmos más complejos, pero a veces no necesitas nada más que una búsqueda simple para encontrar lo que buscas.

Y, por supuesto, el problema del vendedor viajero (TSP) es uno de los problemas más conocidos y desafiantes de la teoría de grafos. Es interesante ver cómo un problema que parece tan simple (encontrar el camino más corto que pase por varias ciudades) se vuelve tan difícil de resolver cuando el número de ciudades crece. Aquí, intentamos resolverlo de manera aproximada con algoritmos heurísticos, ya que la solución exacta no es práctica para grandes entradas.

Lo bueno de este proyecto es que podemos ver cómo cada algoritmo tiene su lugar y cómo se complementan entre sí para resolver diferentes tipos de problemas. A01198201