

En esta evidencia, pude adentrarme en la implementación y comprensión de algoritmos aplicados a grafos, lo cual resultó ser una experiencia enriquecedora y desafiante a la vez. Los grafos, como estructuras fundamentales en muchos problemas informáticos, permiten representar redes de relaciones y conexiones, lo que hace posible la optimización de rutas y la gestión de recursos en una variedad de aplicaciones. Durante este proceso, uno de los principales aspectos que aprendí fue cómo optimizar las rutas construidas a partir de un grafo, considerando los costos generados por cada conexión. Este tipo de optimización es crucial cuando se trata de encontrar soluciones eficientes en situaciones donde los recursos son limitados, como en problemas de transporte, redes de comunicación o incluso en la organización de datos.

Otro aspecto fundamental que abordamos fue el uso de CMake para implementar y ejecutar tests sobre el código desarrollado. Aprendí que CMake es una herramienta esencial para la creación de proyectos y la gestión de la construcción de código, especialmente cuando se trabajan en entornos que requieren una estructura más compleja. La integración de tests es crucial para garantizar que el código funcione de manera correcta y fiable, evitando errores que puedan surgir durante el desarrollo o en fases posteriores. Gracias a esta práctica, pude observar la importancia de probar cada componente del código de manera sistemática, lo que mejora la calidad del producto final y reduce el riesgo de fallos inesperados.

En mi caso, me enfoqué principalmente en el algoritmo de Kruskal, el cual tiene una aplicación directa en problemas de optimización de grafos, especialmente para encontrar el árbol de expansión mínima. Este algoritmo es interesante porque trabaja de manera incremental, añadiendo las aristas más pequeñas a medida que avanza, siempre y cuando no se forme un ciclo en el grafo. Aprendí que este algoritmo es eficiente, especialmente en grafos dispersos, ya que su complejidad es relativamente baja en comparación con otros métodos, lo que lo hace adecuado para grandes volúmenes de datos.

Al implementar el algoritmo de Kruskal, pude profundizar en su funcionamiento, así como en la importancia de la estructura de datos utilizada para su desarrollo, como la estructura de conjuntos disjuntos (union-find). Esta estructura permite gestionar de manera eficiente la unión de conjuntos de nodos, evitando la creación de ciclos mientras se construye el árbol de expansión mínima. Fue fascinante ver cómo el algoritmo se desarrolla paso a paso, y cómo cada decisión tomada impacta en el resultado final.

A lo largo de este proceso, no solo logré entender en profundidad los algoritmos utilizados en grafos, sino también la importancia de contar con herramientas adecuadas para desarrollar, gestionar y probar el código. La experiencia adquirida al trabajar con CMake y realizar pruebas de manera constante me permitió ver cómo la teoría y la práctica se complementan, y cómo un enfoque estructurado y metódico puede llevar a soluciones más robustas y confiables. En resumen, esta evidencia fue una excelente oportunidad para consolidar conocimientos tanto en algoritmos como en herramientas de desarrollo, y me dejó valiosas lecciones sobre la importancia de la optimización y las pruebas en el desarrollo de software.