José Emilio Alvear Cantú
A01024944

Lunes 07 de de noviembre de 2022
Algoritmos Avanzados TC2038.604

Reflexión de Actividad Integradora

Durante esta actividad, aprendimos más acerca de diferentes algoritmos los cuales considero que son más complicados que en la actividad anterior. A mí me tocó la parte del problema 3, el cual se enfocaba en encontrar el máximo servicio que se podía mandar (en otras palabras, el máximo flujo). Para solucionarlo, tuve que usar el algoritmo de Dinic.

Este algoritmo se me hizo muy interesante porque utiliza otros dos algoritmos que ya habíamos visto antes para crear un algoritmo más complejo. Utiliza Breath First Search para crear el grafo auxiliar de niveles, y utiliza Depth First Search para ir calculando el flujo posible. Al principio me costó entender cómo funcionaban los dos algoritmos funcionan juntos para lograr encontrar el máximo flujo, pero cuando logré entenderlo se me hizo muy ingenioso.

También utilizar el grafo de niveles se me hizo una muy buena estrategia porque creo que es una poderosa optimización en este algoritmo. Si no fuera por los niveles, se harían muchas más búsquedas de las que en realidad deben hacerse. Debido a que usamos la matriz de adyacencia, el algoritmo tiene una complejidad de espacio O(V^2) donde V son los vértices. Esto se puede considerar como un aspecto un tanto negativo porque ocupa mucho espacio, pero la ventaja de Dinic está su complejidad de tiempo.

La complejidad de tiempo es de O(EV^2) Esta es una complejidad muy eficiente para un algoritmo de grafos. Lo más relevante es que la complejidad es bastante reducida en cuanto a las aristas. Generalmente, las aristas se recorren a una complejidad mayor, pero la eficiencia del algoritmo demuestra el intelecto de Dinitz, y he aprendido mucho con esta parte de la actividad.

Referencias

GeeksForGeeks. (s.f.). Dinic's Maximum Flow [sitio web]. Recuperado de: https://www.geeksforgeeks.org/dinics-algorithm-maximum-flow/