Adrian Gonzalez Saldivar
A00827845
20/11/2020

Act 4.3 - Actividad Integral de Grafos (Evidencia Competencia)

Como se vio en las clases anteriores un grafo es un conjunto de nodos y de arcos, ambos igualmente son llamados como vértices y aristas respectivamente.Cuando se implementa un grafo se tiene que conocer justamente cuál es el tipo de grafo a utilizar en la situación adecuada, daré dos ejemplos de diferentes grados estos son el grafo dirigido y el no dirigido. Debido a la naturaleza de la situación problema que se nos fue pedida resolver se vio adecuado, la implementación de un grafo dirigido.

En un grafo dirigido los arcos tienen una dirección, a diferencia de un no dirigido que es bidireccional. Un grafo se puede implementar en base de una matriz de adyacencia o de una lista de adyacencia, esta última fue la utilizada en esta situación problema, cada una cuenta con sus ventajas y desventajas. Una de las ventajas del uso de la lista de adyacentes es que hace un buen uso de la memoria y es muy eficiente. Este se utiliza cuando la cantidad de arcos es menor a O(n^2). Su desventaja es en la eficiencia en determinar la existencia de un arco de un vértice a otro, al igual que requiere un mayor espacio de memoria. Este factor fue tomado en cuenta durante el desarrollo de la situación problema, es por eso que se agregó y utilizo la libreria de lo que se conoce como unordered_map, el cual tiene una eficiencia constante (O(1)), lo que significa que hace el proceso mucho más rápido por su implementación.

El algoritmo con mayor complejidad en esta situación problema fue de O(V + E). Lo que esto representa es que la complejidad del algoritmo está basada en la cantidad de vértices y aristas introducidas al programa. La menor complejidad en el algoritmo como antes mencionado es el O(1).