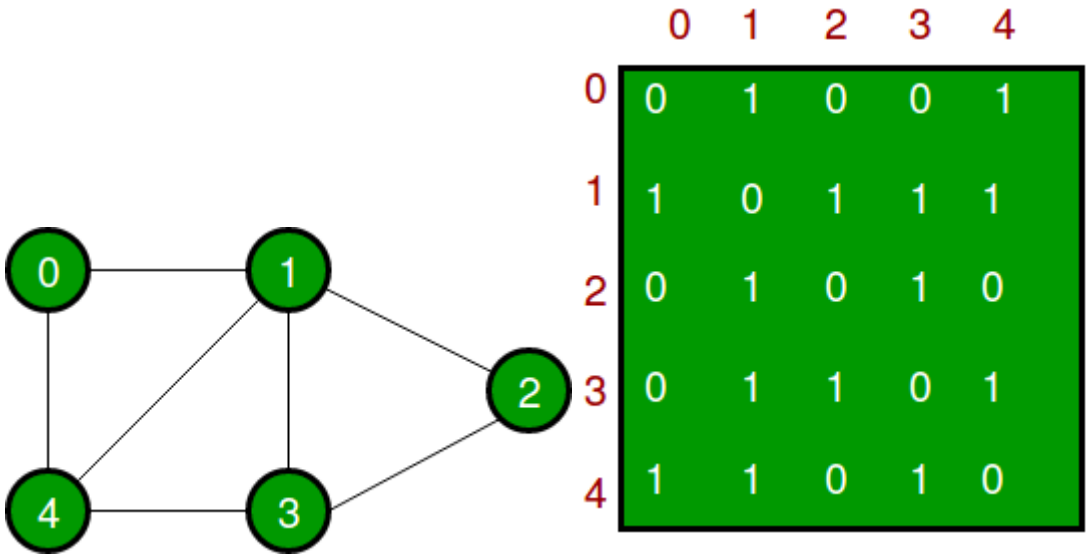
Samuel García Berenfeld A01642317

Reporte de Actividad 4.3

Parte 1: ventajas de nuestra estructura.

Durante la implementación de los métodos pertinentes a la solución de la actividad, utilizamos principalmente la estructura de datos llamada grafo para almacenar nuestros datos. Estas estructuras se definen como un grupo de nodos y edges (que son formados al conectar dos nodos). La implementación particular de nuestro grafo es con la característica de que contiene una lista de adyacencias, que registra las interacciones entre los elementos de la lista. Esto nos permite simplificar nuestros métodos subsecuentes. Esta estructura es ideal para nuestra actividad, ya que una de sus principales ventajas es que permite encontrar distancias más cortas entre vértices de manera rápida (GeeksforGeeks, 2023). Gracias a la implementación de grafos para esta actividad, podemos entonces crear nuestro algoritmo de Dijkstra, ya que podemos consultar a nuestra lista de adyacencias para encontrar los caminos más cortos entre el nodo buscado y los demás en el grafo (aayushi2402, 2023).



Grafo sencillo con su ejemplo de matriz de adyacencia.

Parte 2: explicación de tiempo de algoritmo Dijkstra.

Aquí hablaremos de la complejidad en tiempo del algoritmo de Dijkstra. Para empezar, es importante mencionar que todas las documentaciones de las distintas complejidades están documentadas dentro del código, y pueden ser consultadas ahí. Sin embargo, quiero hablar un poco de como la manera específica en la que creamos nuestro grafo tiene une efecto

importante sobre este algoritmo. Tener una matriz de adyacencias es muy ventajoso para nuestra aplicación, ya que hacerlo de esta manera decrementa la complejidad de este algoritmo específico de O(V^2) donde V es el número de vértices, hasta O(E logV) donde E es el número de edges y V el número de vértices. Esto sucede debido a que si el algoritmo fuera a ser ejecutado en una implementación sin adyacencias tendríamos dos loops de tipo while uno dentro del otro (ambos dependiendo del número de vértices). Sin embargo, ahora tenemos nuestros loops, pero ahora simplemente recorremos nuestra lista de adyacencias, bajando la complejidad (GeeksForGeeks, 2022).

aayushi2402. (2023, January 19). *Applications, advantages and disadvantages of graph*. GeeksforGeeks. Retrieved February 5, 2023, from https://www.geeksforgeeks.org/applications-advantages-and-disadvantages-of-graph/

GeeksforGeeks. (2023, January 18). *Graph and its representations*. GeeksforGeeks. Retrieved February 5, 2023, from https://www.geeksforgeeks.org/graph-and-its-representations/

GeeksforGeeks. (2022, September 22). *Dijkstra's algorithm for adjacency list representation: Greedy Algo-8*. GeeksforGeeks. Retrieved February 5, 2023, from https://www.geeksforgeeks.org/dijkstras-algorithm-for-adjacency-list-representation-greedy-algo-8/