Act 4.3. Actividad Integral de Grafos (Evidencia Competencia)

Alain Vicencio Arizabalo A01620758

Julián Andres Gonzalez Arguello A00831514

Programación de estructura de datos y algoritmos fundamentales

Grupo 570

ITESM

Martes 07 de febrero del 2022

Los grafos son una estructura de datos no lineal compuesta de nodos (también llamados vértices) y arcos (también llamados aristas). El punto de esta estructura de datos es poder almacenar la información de tal manera en que se puedan formar redes. Ejemplos de grafos en la vida real incluyen calles de una ciudad, carreteras entre ciudades, líneas telefónicas, entre otros. En un ámbito digital, los grafos se encuentran en redes sociales como Facebook o Linkedin para poder marcar las conexiones de trabajo o amistades.

(GeeksforGeeks, 2023).

En esta actividad se utilizó una lista ligada para guardar las adyacencias que tuvieran los nodos, es decir, el número de conexiones que tuviera un nodo (que guarda la dirección IP) con otros. Una vez guardado así el grafo, se creaba un Max Heap de objetos tipo Dirección IP para así ser acomodados de mayor a menor según el número de adyacencias que tuvieran los nodos, es aquí donde el nodo con más adyacencias en nombrado como el ‘Boot Master’. Finalmente, a base de un mapeo hecho para obtener los índices de los nodos en la lista de adyacencias, se usa el algoritmo de Dijkstra para encontrar la distancia que tienen los demás nodos del ‘Boot Master’.

Reflexión Alain:

Los grafos como estructura de datos presentan tanto desventajas como ventajas para el usuario. Su peor característica viene siendo el uso y agilidad de memoria con sus implementaciones, esto es visto principalmente con el algoritmo de Dijkstra, que tiene una complejidad de O((|V|+|E|)logn), siendo que depende del número de vértices y de aristas para funcionar. La mayoría de las funciones propias del grafo dependen de una variable,

como lo es findIpIndex para encontrar el índice del nodo (Complejidad O(n)), cargar el grafo para su demostración (Complejidad O(n^2)), gradosIPs para definir la jerarquía que tienen estos nodos en sus conexiones y permite el ordenamiento mediante heap (Complejidad O(n)), o la función cincoMayores que despliega las mayores conexiones a base del heap (Complejidad O(n).

Pero a la vez, eso es lo que brinda su mayor ventaja, que es la formación de una red de información, donde todo nodo está conectado relevantemente con otro. Si dos nodos no están directamente conectados y necesitan de un tercer nodo que los una, los grafos permiten que ese nodo de en medio tenga una relevancia respecto a los otros dos nodos, esto representado por su peso. Su existencia significa algo más allá del que le da el usuario. Esta es la misma lógica con la que Facebook recomienda amigos, une la información a base de otra. Y es con esta misma secuencia con la que se puede demostrar su importancia y efectividad en la situación problema. La infección de una red supone un afecto a los que conectan con él, y así como un bien se puede esparcir por una red, un mal también, pero como el grafo (específicamente con el algoritmo de Dijkstra) puede conocer el mejor camino para llegar a tal nodo, los métodos de ataque se vuelven más eficientes por velocidad. La mayor ventaja de los grafos es su velocidad para llegar a al destino y el peso que el camino tiene.

Referencias:

Sin autor. (2023). Graph Data Structure And Algorithms. GeeksforGeeks. Recuperado de https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/