

Estructura de datos y algoritmos fundamentales

Víctor Adrián Sosa Hernández

*Actividad integradora 4.1*

***Reto***

**María Fernanda Hernández Montes - A01704918**

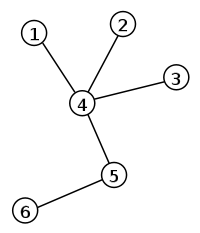
**Paulina Cardoso Fuentes - A01701490**

**Paola Adriana Millares Forno - A01705674**

Campus Querétaro

05 de febrero de 2021

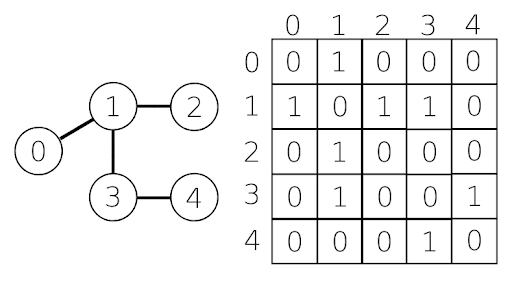
***GRAFOS***

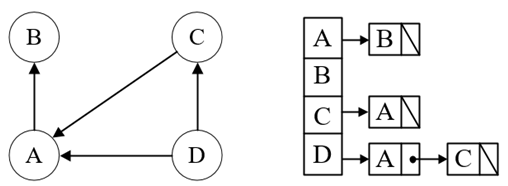
Se trata de un tipo abstracto de datos que se encuentra compuesto por dos elementos principales: nodos y arcos. Mientras que los nodos son los elementos encargados de almacenar la información en cuestión, los arcos representan aquellas conexiones que permiten viajar de un nodo a otro.

De esta manera, los grafos nos brindan diversos beneficios al momento de trabajar con el almacenamiento y acceso a la información, pues nos permiten representar redes que serán la manera de brindar soluciones a estructuras o proyectos tales como la identificación de trayectorias en un espacio geográfico o la conexión entre usuarios de aplicaciones como Reddit, Twitter, Facebook, entre otras.

Existen dos maneras comunes de trabajar con la representación de los grafos, como podemos ver a continuación.

**Matriz de adyacencia**

Este tipo de representación para grafos nos permite conocer la existencia de arcos entre dos nodos, de manera que su tamaño es VxV donde V representa el número de nodos existentes. Cuando una posición dentro de la matriz almacena un 1 esto significa que, en efecto, existe un arco conectando ambos nodos adyacentes; mientras que si se almacena un 0, no hay una conexión entre los elementos.

**Lista de adyacencia**

Para este tipo de representación implementamos un arreglo de listas, donde el tamaño del arreglo es la cantidad de nodos con los que cuenta el grafo, y los elementos al interior de la lista representan los nodos conectados con el nodo en cuestión en dicha posición del arreglo.

Para recorrer esta estructura de datos existen dos tipos de algoritmos que nos permitirán realizar búsquedas entre todos los nodos: Breadth First Search (BFS) y Depth First Search (DFS), que veremos más adelante.

**Breadth First Search (BFS)**

Básicamente, este algoritmo se encarga de recorrer el grafo cuidando evitar el pasar más de una ocasión por un mismo nodo, en caso de que exista un bucle en el que un elemento se apunte a sí mismo. De esta manera, el recorrido en anchura consiste de las siguientes reglas:

* Se visita el nodo adyacente al nodo inicial y se marca como visitado, se muestra y después se inserta en una fila;
* Si no existen nodos adyacentes se elimina el nodo inicial de la fila;
* Se ejecutan los puntos anteriores de manera repetida hasta que la fila se encuentre vacía.

**Depth First Search (DFS)**

Al igual que el algoritmo para el Breadth First Search, este algoritmo cuida la existencia de bucles en los nodos que cuentan con arcos apuntando a sí mismos, con la diferencia de que en este caso trabajaremos con una pila.

Básicamente, para la ejecución de este algoritmo tomamos un nodo de referencia, que será el primero en entrar en la pila. Posteriormente nos moveremos al nodo adyacente de menor valor e igualmente lo añadimos a la pila. Esto se mantiene sucediendo hasta que el nodo en el que nos encontramos ya no tiene nodos adyacentes no visitados; cuando se da este caso, el nodo se elimina de la pila (proceso que igualmente se repite hasta vaciarla).

***Reflexiones individuales:***

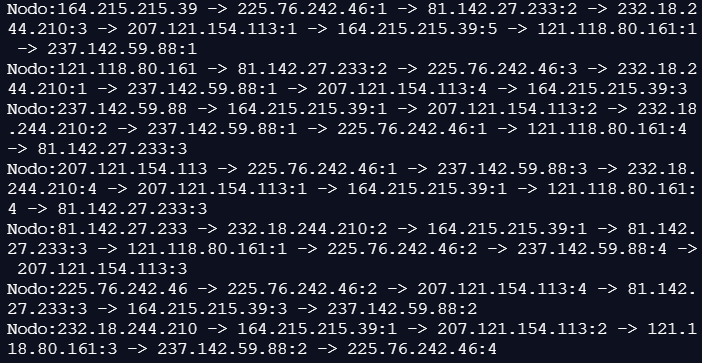
***Fernanda:***Este avance de la situación problema me ha permitido identificar la versatilidad que brindan los grafos al momento de manejar conjuntos grandes de información, permitiéndonos reducir el tiempo y la complejidad implicada al momento de llevar a cabo estos procesos por medio de otros tipos de estructuras. Durante esta actividad fui capaz de identificar algunos puntos débiles que tengo con respecto a esta estructura de datos en particular, y, por ende, conocer igualmente las maneras adecuadas a mi estilo de aprendizaje para mejorar mi capacidad y agilidad al momento de implementarlos en futuros proyectos concernientes al área.

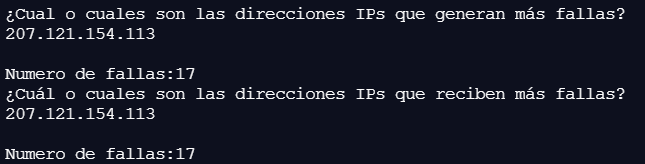
La actividad y el intentar resolverla, tanto en lo individual como en equipo, me permitió comprender las principales dificultades y puntos clave de los grafos como herramienta para la solución a problemas.

***Paulina:*** Al realizar esta actividad, logré entender mucho mejor el manejo de grafos, cómo interactúan los nodos entre ellos y toda la lógica e implementación que se implica dentro de esta estructura de datos. Me fue de gran utilidad realizarla pues pude ver una situación cotidiana en la que se pueden utilizar los grafos y cómo éstos nos ayudan a manejar la información de una forma más simple. La actividad me ayudó a despejar dudas y poner a prueba los conocimientos que vimos en clase, de manera en la que pude fortalecer lo que aprendí durante las clases. También creo que fue un reto pues a veces con tantas conexiones entre nodos y mucha información, y al hacer uso de muchos apuntadores, de vez en cuando me confundía con eso. Sin embargo, al final pude analizar todo hasta entender perfecto todo el funcionamiento del programa.

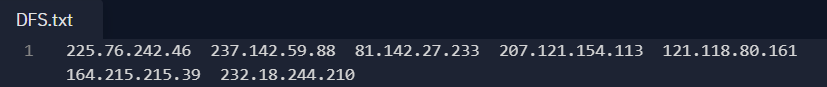
***Adriana:***Este trabajo me dio la oportunidad de practicar y afianzar los conocimientos que fui adquiriendo hasta este punto del curso. La lógica empleada fue uno de los mayores desafíos, ya que el uso de apuntadores y registros de memoria a veces puede ser algo complicado, pero una vez que conoces tu código y lo entiendes, esto se vuelve algo simple que te ayuda a hacer un trabajo de manera más eficiente. El uso de distintas funciones y la investigación que debimos realizar para resolver este problema, fueron de las cosas más enriquecedoras, ya que nos da la certeza de que podemos obtener aptitudes distintas y demuestra que vamos adquiriendo la capacidad de adaptarnos a distintas problemáticas. El uso de las distitas estructuras de datos, dentro de las listas, me permitió idear formas de abordar los problemas de distintas maneras, lo cual me hace pensar en formas de optimizar el código y la lógica que hay por detras. El uso de grafos siempre es algo retador, ya que el esquematizar e imaginar la estructura puede ser algo complicado, el uso de las funciones, aunque fue algo que me resulto dificil en un principio, pude entenderlas e implementarlas.

***Casos de prueba:***

******

******

******

******

***Referencias bibliográficas:***

Guru99. (Desconocido). BFS vs DFS: Know the difference. Febrero 2021, de Guru99 Sitio web: <https://www.guru99.com/difference-between-bfs-and-dfs.html>

GeeksforGeeks. (Diciembre 2020). Breadth First Search or BFS for a Graph. Febrero 2021, de GeeksforGeeks.org Sitio web: <https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/>