Actividad 5.2: Reflexión individual

Estudiante: Diego Palma Rodríguez **Matrícula:** A01759772

La situación problema consiste en analizar y manipular registros almacenados en bitácoras, los cuales poseen información sobre los intentos de acceder a una red. Los registros de la bitácora contienen el mes, día, hora, IP origen, IP destino, costo y el estatus del intento de acceso. En este caso, se solicita almacenar las IPs en un grafo; y a partir de ello, generar una tabla Hash en la que se almacene un resumen de la información de todas las IP: IP, total de direcciones accesadas por la IP y total de direcciones que intentaron acceder a la IP. Además de ello, se pide determinar la cantidad de colisiones al ingresar la información a la tabla Hash. Finalmente, se debe proporcionar al usuario la información de la IP que solicite, verificando si esta se encuentra en la tabla Hash.

Por un lado, es importante definir la estructura de datos que se utilizará para almacenar los registros. En este caso se utilizará una lista de adyacencias (utilizada en actividades anteriores) para representar el grafo. Un grafo consiste en un conjunto de vértices y arcos [1]. Cada arco consiste de un par (v, w), pero en este caso en particular presentará una tercera componente que almacena el costo entre ambos vértices. De este modo, cada IP estará almacenada en cada nodo y se pueda determinar cuántos accesos realizó al calcular su grado. Además de ello, se creó una estructura de datos tipo *map* para almacenar la cantidad de direcciones que accedieron a la IP. Esto resulta más eficiente que solo almacenar un grafo y luego utilizar un algoritmo para contar la cantidad de direcciones.

Por otro lado, la implementación de la tabla Hash es comúnmente llamada *Hashing*. *Hashing* es una técnica utilizada para realizar operaciones de inserción, eliminación, y búsqueda en un tiempo constante [1]. En la presente actividad se usó el método de prueba cuadrática para la inserción de datos, lo cual no suele ser eficiente en ciertos casos, pero resulta sencilla de implementar. Con respecto a la función para obtener el índice hash de cada llave, está consistió en sumar los tres primeros números de la IP y se multiplicó por el cuarto número; y luego se obtuvo el módulo del resultado en base al número primo mayor y más cercano a la cantidad de IPs. Dicha función se obtuvo en base a prueba y error hasta obtener una función con la menor cantidad de colisiones. Finalmente, la función para encontrar cierto valor en la tabla Hash consiste en obtener su valor hash y acceder directamente a su contenido. Esto resulta muy eficiente en comparación a otras estructuras de datos.

REFERENCIAS

[1] Weiss, M. A. (2014). Data structures & algorithm analysis in C++. Reading, Mass: Addison Wesley.