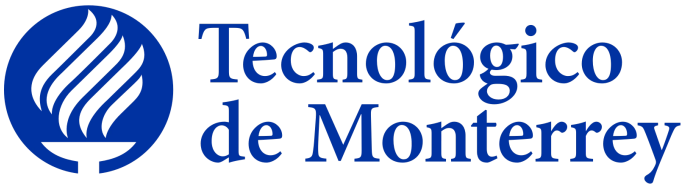
Instituto Tecnológico y de Estudios Superiores de Monterrey
Escuela de Ingeniería y Ciencias



Programación de Estructuras de Datos y Algoritmos Fundamentales
Grupo 601

Actividad 5.2
Actividad Integral de Códigos Hash

Natalia Sofía Salgado García
A01571008

11 de junio del 2023

Reflexión

Las tablas hash son estructuras de datos en donde se almacenan pares: una llave, la cual es un dato único que no se puede repetir en la tabla, y un dato asociado a esa llave. Esta estructura de datos es particularmente útil en casos donde se requiera almacenar una sola instancia de cada dato en un conjunto de datos.

Para almacenar los datos, la tabla hace uso de funciones hash, en donde se toma como parámetro la llave para que se realice algún tipo de procesamiento a partir de ella y con eso se obtenga el índice donde se debería colocar la llave en la tabla. Estas tablas son especialmente eficientes en cuanto a búsqueda de valores, ya que, como se puede inferir, al buscar una llave en la tabla, la complejidad de tiempo es, en el mejor caso, O(1), pues al buscar la llave, ya se sabe en qué índice debe encontrarse debido al hashing. No obstante, pueden ocurrir colisiones que hagan que la complejidad incremente, dado que, al estar un índice ocupado, la llave debe ser guardada en otro índice (hablando del método lineal), lo cual ahora requerirá iteraciones para encontrar dicha llave en la tabla.

Para esta actividad, se pidió guardar cada IP (sin contar el puerto) en una tabla hash y que cada registro se guardara de acuerdo a su IP. Una tabla hash es útil para este fin, ya que la IP actúa como llave única, por lo que no se puede repetir. La actividad también pedía que al buscar una IP, se desplegaran todos sus registros. Debido a que es básicamente una actividad basada en búsqueda, esta estructura de datos es una muy eficiente, pues su complejidad de tiempo tiene el potencial de ser mucho menor que el de otros tipos de búsqueda, como secuencial con complejidad de O(n) o incluso binaria con O(log n).

Para realizar la actividad, se implementó una tabla hash donde se manejan las colisiones con encadenamiento, es decir, si alguna llave otorga un índice ya ocupado al aplicar el hashing, la segunda llave se encadena a la previa en ese índice, por lo que se crean listas encadenadas. La búsqueda con este método tiene igualmente una complejidad de O(1) en el mejor caso (en donde la llave sea la cabeza de la lista); no obstante, si hay varias llaves en un mismo índice, la complejidad aumenta a O(n), donde n es el número de llaves en el índice. En cuanto a la inserción, de la misma manera tendría una complejidad de O(n) donde n es también el tamaño de la lista, ya que es necesario primero ver si la llave existe en la tabla, y en caso de que no, recorrer la lista en el índice para colocarla al final.

En general, se espera que las listas encadenadas para cada índice sean cortas, ya que, de lo contrario, la estructura de datos pierde la eficiencia en la que se basa.

Link al video:

https://youtu.be/TiN-7ejI_E4