Instituto Tecnológico y de Estudios superiores de Monterrey

Programación de estructuras de datos y algoritmos fundamentales

5.2 - Actividad Integral sobre el uso de códigos hash

Salvador Alejandro Gaytán Ibáñez A01730311

28 de noviembre de 2020

Puebla, Pue.

Reflexión

Hash Tables:

Una Hash Table (tabla hash, mapa hash, table de dispersión, tabla fragmentada o matriz asociativa), es una estructura de datos que permite al programador almacenar keys (llaves) las cuales son únicas y se le asignan a cada uno de los datos a almacenar y valores dada una colección de elementos.

Las Hash Tables se utilizan cuando existe la necesidad de encontrar uno de los elementos de un conjunto de datos rápidamente mediante una función aritmética.

Desde el punto de vista computacional, una Hash Table se define como un espacio en memoria contigua subdividida en cajones o buckets que generalmente se implementas con un arreglo y listas enlazadas.

La manera común de realizar la función de Hashing consiste en dos pasos:

1) Un elemento es convertido en un entero usando una función de Hash, el elemento puede ser usado como un índice para almacenar el elemento original que se quiere guardar en la Hash Table.
2) El elemento se guarda en la Hash Table donde puede ser accedidos de manera rápida y eficiente usando su llave única (hashed key).
   Pseudocódigo:
   Hash= hashfunc(llave)
   Índice=hash%tamaño_tabla.

Una de las funciones más útiles de las Hash Tables se le conoce como Hashing function y consiste en identificar de forma única un objeto específico de un grupo de objetos similares. Algunos ejemplos de cómo se usa la Hashing function en la vida cotidiana es en las instituciones donde a cada individuo miembro de ella se le asigna un identificador único (en el caso del Tecnológico de Monterrey, la nómina o matrícula) que se puede utilizar para recuperar información sobre ellos, siendo que no existen dentro de la organización 2 identificadores iguales.

La Hashing Function es cualquier función que puede ser utilizada para mapear un conjunto de datos de tamaño arbitrario a un conjunto de datos de tamaño fijo que recae dentro la Tabla Hash.

Para poder hacer un buen mecanismo de hasheo se recomienda que la Hashing Function cumpla con los siguientes requerimientos básicos:

- Debe de ser fácil de procesar y no se debe de convertir en su propio algoritmo
- Debe de proveer una distribución uniforme a lo largo de toda la tabla y evitar que se conglomere información en lugares de la tabla

- Reducir en medida de lo posible las colisiones, es decir, que dos elementos sean mapeados en el mismo lugar dentro de la Hash Table.

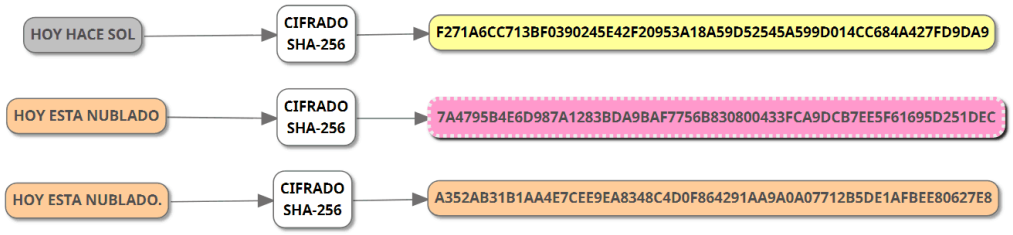
La mayor ventaja del uso de una Hash Table es que los elementos almacenados dentro de la misma pueden ser accedidos de manera más rápida a diferencia de otras estructuras de datos, el problema es que debido a la Hashing Function, no se tiene control sobre el orden de los elementos contenidos dentro de la Hash Table ya que la misma estructura de datos los organiza según un orden que le conviene para poder recuperar rápidamente los elementos de esta.

Las Hash Tables son estructuras de datos muy útiles para acceder información de manera rápida donde no es tan importante el no tener control sobre la manera en que se ordenan, lo cual no es tan importante para la situación problema resuelta ya que lo más fundamental es hallar la información, no importa la manera en la que esta sea guardada.

SHA-256:

El SHA-256 es un algoritmo utilizado por Bitcoin para garantizar la integridad de la información almacenada en un bloque, su nombre, SHA, viene del inglés Secure Hash Algorithm (Algoritmo de Hasheo Seguro) y es uno de los avances mundiales en materia de criptografía que fue impulsado debido a las guerras, fue desarrollado en conjunto por la Agencia de Seguridad Nacional de los Estados Unidos y por el Instituto Nacional de Estándares y Tecnología con el objetivo de generar códigos o hashes únicos en base a un estándar que permitiera asegurar datos o documentos frente a cualquier agente externo que quiera modificarlos.

El protocolo SHA se hizo público por primera vez en 1993 y fue conocido como SHA-0, después salió el SHA-1, SHA-2 y posteriormente otras variantes como la SHA-224, SHA-256, SHA-384 y SHA-512 donde lo que diferencia a cada variante es el número de bits. De todos estos algoritmos para crear hashes, el SHA-256 es uno de los más populares debido a su equilibrio ente coste computacional de generación y seguridad ya que es un algoritmo extremadamente eficiente para la alta resistencia a colisiones que tiene. Una de las particularidades que también aportan a que el algoritmo SHA-256 sea tan utilizado es que la longitud de hash resultante después de la Hashing Function siempre tiene la misma longitud, una cadena de 40 letras y números con una codificación de 32 bytes o 256 bits sin importar la longitud del dato a cifrar.



Ventajas:

- La llave de 256 bits es más segura que las generadas por otros algoritmos de hashing.
- Las colisiones son muy raras.
- Cuenta con el efecto avalancha el cual consiste en que inclusive con el menor cambio a la información original los valores hash se cambian completamente.

Desventajas:

- Alto uso de recursos computacionales cuando la seguridad de los datos no es prioridad.
- Al ser un hash de encriptación, puedes darle el valor y obtener la llave, pero nunca puedes dar la llave y obtener el valor

El SHA-256 es usado en algunas de los protocolos mas populares de autenticación y encriptación como lo son SSL, TLS, IPsec, SSH y PGP, asimismo, en Linux y Unix, se usa para hashing de contraseñas seguro y finalmente también es empleado en criptomonedas como la Bitcoin para verificar transacciones.

Complejidad: O(c+xn) donde n es el numero de bloques requeridos para ajustar el input, x es la constante para sobrecarga por bloque y c es la constante para la inicialización y finalización.

El SHA-256 es un algoritmo muy poderoso de hasheo y si nuestra situación problema fuera más que eso y realmente manejara datos sensibles sería una muy buena manera de mantenerlos seguros, sin embargo, debido a que es una tarea escolar el uso del SHA-256 sería bastante overkill.

Referencias:

Garg, P. (N/A). Basics of Hash Tables . 28/11/2020, de Hacker Earth Sitio web: https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/tutorial/

Zavaleta, R. (2016). Asi funcionan los Hash Tables. 28/11/2020, de Bit y Byte Sitio web: http://bitybyte.github.io/Hashtables/

N/A. (N/A). ¿Qué es SHA-256?. 28/11/2020, de bit2me academy Sitio web: https://academy.bit2me.com/sha256-algoritmo-bitcoin/

Natanael. (2019). What is the time complexity of computing a cryptographic hash function/random oracle?. 28/11/2020, de Stack Exchange Sitio web: https://crypto.stackexchange.com/questions/67448/what-is-the-time-complexity-of-computing-a-cryptographic-hash-function-random-or

SolarWinds MSP. (2019). SHA-256 Algorithm Overview . 28/11/2020, de SolarWinds MSP Sitio web: https://www.solarwindsmsp.com/blog/sha-256-encryption