Actividad 5.2 – Actividad Integral sobre el uso de códigos hash

Mauricio Cantú Torres A01633805

Joel Isaías Solano Ocampo A01639289

Programación de estructuras de datos y algoritmos fundamentales

Grupo 12

Instituto Tecnológico y de Estudios Superiores de Monterrey

Sábado 4 de diciembre de 2021

Actividad 5.2 – Actividad Integral sobre el uso de códigos hash

En esta actividad, se solicita que utilicemos una tabla hash para almacenar un resumen de la información de las IP's en la bitácora. La bitácora se compone por 13370 registros, por lo que cuestiones de búsqueda e iteración de los elementos tomaría mucho tiempo si se usara una estructura de datos diferente a una tabla hash. La ventaja de utilizar la tabla hash es que ofrece la posibilidad de insertar, borrar y consultar elementos con una complejidad temporal constante, por lo cual utilizar esta estructura es ideal para trabajar con una bitácora que contiene tantos registros.

Tablas hash

Las tablas hash son estructuras de datos que se utilizan para implementar un arreglo asociativo, esto quiere decir que se puede asignar una llave a cada valor de la tabla, esta es la clave detrás de la complejidad temporal tan notoria de las tablas hash. Sin embargo, otro elemento crucial de las tablas hash es el método hashing, el cual se utiliza para determinar en qué parte de la tabla se debería colocar un valor, el cual, después de ser insertado, puede ser buscado o eliminado de la tabla.

Complejidad temporal

A pesar de que las tablas hash son famosas por su complejidad temporal constante en el proceso de inserción, eliminación y búsqueda, métodos cuya complejidad es O(1). Hay una asunción que invalida lo anteriormente planteado, el lugar de la tabla hash en el que se está intentando insertar un valor no está ocupado. Cuando se intenta insertar un valor en un lugar ocupado de una tabla hash, se le denomina "colisión". Existen diferentes formas de lidiar con colisiones, como, por ejemplo, métodos hash "perfectos", que evitan que existan colisiones en primer lugar, sin embargo, este método es mucho más complejo que sus alternativas. Uno de los métodos de hashing más conocidos (y el utilizado en clase) es el hashing abierto, el cual consiste en repetir el proceso de hashing cuando se detecta una colisión una y otra vez hasta encontrar un lugar desocupado. También, se puede optar por el método de chaining, el cual consiste en asignar múltiples valores a un mismo lugar en una tabla hash, lo cual ocurre únicamente cuando se detectan colisiones. Ambos métodos son utilizados para lidiar con colisiones, sin embargo, cuando las colisiones incrementan, surgen dos problemas que aumentan la complejidad temporal de la tabla. El primero consiste en el rehashing, este método se lleva a cabo cuando se excede el balance de carga de una tabla hash y se incrementa el tamaño de la misma para poder alojar más elementos. El segundo método es aplicado cuando se inserta más de un elemento en un espacio de la tabla, ya que, para lograr esto, se utilizan listas ligadas. A pesar de que esto permite alojar más de un elemento en un mismo lugar, la complejidad temporal aumentaría

inevitablemente en estos casos, porque ahora se debe verificar cada elemento de cada espacio de la tabla cuando se realiza una búsqueda. En conclusión, es impredecible saber el número de colisiones que se tendrán al utilizar una tabla hash, y, a pesar de que la complejidad temporal de esta es, presumiblemente, constante en la mayoría de los casos, es la necesidad de lidiar con colisiones lo que, potencialmente, podría incrementar la complejidad temporal, dependiendo del número de colisiones que se presenten.

Referencias

Garg, P. (s.f.). Basics of Hash Tables. Recuperado de

https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-

tables/tutorial/