



Tecnológico de Monterrey

Luis Enrique Gutierrez

A00837481

Programación de Estructura de Datos y algoritmos Fundamentales (Gpo 850)

Profesor:

Dr. Eduardo Rodriguez Tello

8 de Junio del 2024

Reflexión Actividad Integradora 5

Tablas Hash

Uso de Tablas Hash

Las tablas hash son otro tipo de estructura de datos que se basan en almacenar y asociar todo tipo de información. Es decir cada dato almacenado en la tabla tendrá un número (clave) única para poder ser accedido. Se caracteriza por ser una estructura sumamente eficiente a la hora de recopilar datos previamente almacenados. Como consiguiente, se podrán hacer una serie de operaciones como: agregar, buscar y remover elementos de la tabla. Para comprender de una manera más sencilla esta estructura, podemos relacionarla a otra estructura de carácter lineal como lo son los arreglos. En cuanto a estructura ambos tipos son muy similares, lo cual favorece a poder entender el funcionamiento de una tabla hash.



Jorge	0
Raúl	1
María	2
Marcos	3
Martín	4
Sergio	5

Ejemplo de una tabla hash

Imagen Obtenida de: [Link](#)

En cuanto a las complejidades de las funciones de las tablas hash, al realizar una pequeña investigación se descubrió que en promedio las operaciones de esta estructura de datos es de $O(1)$ o en el peor de los casos complejidad $O(n)$. Es decir que tiene una serie de algoritmos que se ejecutan de manera rápida. Esto demuestra que las tablas hash son un método confiable para almacenar grandes cantidades de información y que esta se pueda acceder fácilmente.

Relacionando lo anterior a los métodos empleados en el proyecto tienen las siguientes complejidades:

Método	Complejidad
getIndex & getNumElements	$O(1)$
add	$O(1)$
find	$O(1)$
print	$O(n)$
remove	$O(1)$

Tabla 1: Complejidades computacional de métodos

Las tablas hash demostraron ser un gran aliado para manejar grandes cantidades de información, en este caso las Ips. La única desventaja vista es que su desempeño se puede ver afectado por colisiones, que es cuando dos elementos tienen el mismo valor dentro de la tabla. Esto se puede evitar al implementar métodos eficientes que reduzcan el valor de carga que es lo que le da una clave a cada elemento de la tabla. Si se logra lo anterior, se lograra un proyecto exitoso donde las colisiones de elementos no afectarán el desempeño rápido y eficiente de nuestra estructura de datos.

Referencias

Datastructures tool. DSTool: Herramienta para la programación con estructuras de datos. (n.d.). <https://www.hci.uniovi.es/Products/DSTool/hash/hash-queSon.html>

GeeksforGeeks. (2024a, May 8). *Hash table data structure*.
<https://www.geeksforgeeks.org/hash-table-data-structure/>

GeeksforGeeks. (2024b, May 8). *Hash table data structure*.
<https://www.geeksforgeeks.org/hash-table-data-structure/>

Mishra, S., & Ezzat, M. (2024, January 31). *How do you handle hash table collisions?*.
How to Handle Hash Table Collisions in Programming.
<https://www.linkedin.com/advice/0/how-do-you-handle-hash-table-collisions-skills-programming#:~:text=One%20way%20to%20avoid%20collisions,full%20the%20hash%20table%20is> .

University of Missouri–St. Louis. (n.d.).
https://www.umsl.edu/~siegelj/information_theory/projects/HashingFunctionsInCryptography.html#:~:text=Definition%3A%20A%20collision%20occurs%20when,generated%20by%20the%20hash%20function .