



# Tecnológico de Monterrey

## **Act 4.3 - Actividad integral sobre el uso de códigos Hash (Evidencia competencia)**

### **Reflexión**

Alan Paul García Rosales A01639631

Estefanía Pérez Yeo A01639270

Viernes, 3 de diciembre del 2021

Programación de estructuras de datos y algoritmos fundamentales

TC1031 Gpo. 12

Dr. Eduardo Arturo Rodríguez Tello

## **Introducción**

La situación a resolver para esta actividad es algo distinta, en comparación a las actividades anteriormente realizadas, en esta ocasión se nos entrega un archivo de texto que contiene más del doble de registros, pero para esta nueva entrega se solicita que se organicen mediante el uso de un código Hash apoyándonos en el conocimiento adquirido a lo largo del tercer periodo como lo son los Grafos, el cual nos ayudará a mostrar las conexiones entre las ips, además de las ips que han sido accedidas desde cierta dirección como las que han accedido a la misma.

## **Respecto al algoritmo realizado**

Al igual que la vez anterior, a nuestro parecer, la mejor manera de abordar la problemática era crear una clase donde pudiéramos guardar los registros, para luego poder usar algún algoritmo de ordenamiento que fuera lo suficientemente eficiente como para ordenar más de 100,000 registros en total, con un tiempo relativamente corto utilizando un archivo de texto que nos fue proporcionado, en esta ocasión, a la hora de abrir el archivo, decidimos usar getlines individuales, para ver de mejor manera como sucede la lectura del archivo, además, hemos decidido que cada octeto de la ip fuera una variable, no un arreglo de tamaño 4 como en la ocasión anterior..

Posterior a la realización de nuestra clase registro y generar un grafo dirigido que contuviera todos los registros de la bitácora que se nos entregó, procedimos a meter la información de cada una de las ip en una tabla hash, entre la información contenida se encuentra la IP, las IP que intentaron acceder a ella y las que dicha IP intentó acceder.

Finalmente, se nos solicitó identificar las colisiones que sucedieron mientras se ingresaban los datos a la tabla Hash, así como permitir hacer una búsqueda al usuario en consola, mediante el uso de búsqueda binaria esto fue posible y se muestra en consola cada que se compila y corre el código.

## Respecto a la importancia y eficiencia del uso de códigos Hash

Consideramos, que a diferencia de la actividad pasada, en esta ocasión el código compiló y corrió de una manera significativamente más rápida, por lo que, nos podemos dar cuenta que, los códigos hash eficientan de manera importante el manejo de almacenamiento de información.

Puesto que el archivo utilizado con bitácora dentro de la evidencia, es relativamente larga en cuestión de número de IPs, es importante encontrar y manejar de la forma más eficiente posible los datos, y como pudimos notar, los códigos hash, funcionan de una manera significativamente más eficiente que los códigos que hemos utilizado anteriormente.

## Complejidad de los métodos utilizados

A continuación se muestra la complejidad de los métodos necesarios:

CLASE	MÉTODO <sup>1</sup>	COMPLEJIDAD
Graph.h	loadGraphList( );	O(n)
	heapSort();	O(nlogn)
	heapify();	O(n)
	getAdjList();	O(1)
	binarySearch( );	O(n)
	Graph( )	O(1)
	~Graph(	O(1)
	printGraph( )	O(n <sup>2</sup> )

---

<sup>1</sup> Los métodos descritos en la tabla no contienen los argumentos necesarios, solo son para visualizar su complejidad.

IP.h	sobrecarga de operadores >, <, ==, !=, <=, >=, <<	O(1)
	IP( )	O(1)
	getSummary( )	O(1)
	getDecimal( )	O(1)
	updateSummaryDepartures( )	O(1)
	updateSummaryEntrances( )	O(1)
	setAdjIP( )	O(1)
	getIPvect( )	O(1)
Summary.h	Summary()	O(1)
	getIP()	O(1)
	getEntrances()	O(1)
	getDepartures( )	O(1)
	printSummary( )	O(1)
	updateEntrances( )	O(1)
	updateDepartures( )	O(1)
HashTable.h	HashTable()	O(1)
	fill()	O(n)
	getIPSummary()	O(n)
	getInput()	O(n)
	getTable()	O(1)
	getHashIndex()	O(1)
	getSize()	O(1)
	getCol()	O(1)

	print()	$O(n)$
	add()	$O(n)$
	find()	$O(n)$
	getDataAt()	$O(n)$
	remove()	$O(n)$
	quicksort()	$O(n \log n)$
	partition()	$O(n)$
	swap()	$O(1)$
HashNode.h	Todos los métodos	$O(1)$

## Conclusión

Al igual que en la actividad integral anterior, podemos decir que la programación brinda una gran variedad de alternativas para poder dar solución a problemas de la vida cotidiana, aunque en realidad al visualizar las conexiones de más de 105,280 registros resulta complejo este se logra de manera factible gracias a la programación.

Ahora, centrándonos más en la parte del código, respecto a los grafos, como ya comentamos en el párrafo anterior, nos son de mucha utilidad para poder visualizar las relaciones que hay entre distintos objetos dentro del mismo, además, obtener conclusiones de estas conexiones que pueden ser de mucha utilidad.

También consideramos que algoritmos como los utilizados para el hashing son de bastante utilidad, conociéndolos y sabiéndolos aplicar, podríamos almacenar un sin fin de registros de manera eficiente y segura, pudiendo acceder a ellos en cualquier momento de manera rápida.

Para esta quinta entrega, podemos añadir que, las estructuras de datos, como lo son los códigos hash, nos facilitaron de manera sustancial el trabajo, a comparación de otras

estructuras de datos, el objetivo de comparación de conexiones de IPs se pudo realizar con mucha facilidad y rapidez que solo con el heap sort y los grafos. Con esto se concluye que, el uso de las estructuras de datos que aprendimos este tercer periodo, nos facilita el trabajo, si es oportuno utilizarlas.

Se cierra comentando que la presente actividad integradora nos ayudó a darnos cuenta de la gran utilidad de las estructuras de datos que hemos estado aprendiendo a lo largo del semestre de agosto a diciembre 2021 y una vez más, ver que el alcance de la programación no tiene límites, los límites los ponemos nosotros.

### **Extra**

[Link a replit](#)

### **Bibliografía<sup>2</sup>**

GeeksforGeeks. (2021, septiembre 15). *HeapSort*. Recuperado 6 de noviembre de 2021, de <https://www.geeksforgeeks.org/heap-sort/>

---

<sup>2</sup> Las siguientes citas no solo nos fueron informativas, también fueron de ayuda a la hora de implementar código y de algunas de ellas se referenció código directamente y se adaptó a la evidencia.