

## **David Alejandro Lozano Arreola A01722728**

### Programación de estructuras de datos y algoritmos fundamentales

#### Evidencia 4 - Reflexión Final

Durante el desarrollo de las evidencias, fui capaz de implementar distintos algoritmos y estructuras de datos. Pasando desde simples listas, a listas enlazadas, árboles y grafos. Cada una de ellas tiene sus ventajas y desventajas, sin embargo, es necesario mencionar que no existe un algoritmo o una estructura de datos perfecta. La efectividad de cada una depende de los requerimientos del problema a resolver. Es por eso que es necesario analizar bien el problema para poder implementar el mejor algoritmo o estructura de datos. La manera más sencilla de identificar la eficiencia de estas soluciones es a través de funciones matemáticas que predicen el comportamiento de las diversas funciones de las estructuras de datos.

En el caso de la evidencia, las principales operaciones fueron la inserción de elementos, el ordenamiento y la búsqueda de datos. Por lo cual, una manera de destacar la estructura de datos más eficiente es observar el tiempo que tardan en realizarse esas tres operaciones. En el caso del arreglo, tanto la inserción como la búsqueda tienen complejidad  $O(n)$ . Lo cual quiere decir que en el peor de los casos, se requieren “n” operaciones para encontrar un dato en un arreglo de “n” elementos. Sin embargo, el uso de un BST (Binary search tree) nos daría la oportunidad de encontrar un elemento en tiempo  $O(\log(n))$ . Además, una hash table nos permite encontrarlo en  $O(1)$ . Dichas mejoras se crean a partir de la manera en que están organizadas las estructuras de datos. Por ejemplo, el BST nos permite descartar la mitad del arreglo en cada comparación, por lo cual, su búsqueda se comporta de forma logarítmica. Sin embargo, este no es tan rápido como la hashtable, la cual encuentra un elemento en tiempo constante debido a su hash function.

En resumen, cada estructura de datos tiene sus ventajas y desventajas. Además, hay veces en que la simplicidad y mantenibilidad del código son más importantes que la eficiencia. Es por esto que a pesar de ver una mayor eficiencia en las listas enlazadas, considero favorable el uso de arreglos debido a que la cantidad de datos no es tan grande. Sin embargo, considero que las estructuras como el BST y los grafos son mejores que los anteriores por las características que contienen. Entonces, una manera de mejorar las soluciones de cada evidencia, es utilizar la estructura de datos cuyas características mejor se adapten al problema. Por ejemplo, utilizar grafos cuando se busca representar relaciones entre entidades, o utilizar hashtables cuando se quiere optimizar la búsqueda.

## Referencias

*Binary Search Tree*. (2023). Programiz.com.

<https://www.programiz.com/dsa/binary-search-tree>

*Graph Data Structure*. (2023). Programiz.com.

<https://www.programiz.com/dsa/graph>

*Hash Table Data Structure*. (2023). Programiz.com.

<https://www.programiz.com/dsa/hash-table>

*Linked List Data Structure*. (2023). Programiz.com.

<https://www.programiz.com/dsa/linked-list>

*Time complexities of different data structures*. (2020, December 29).

GeeksforGeeks; GeeksforGeeks.

<https://www.geeksforgeeks.org/time-complexities-of-different-data-structures/>