

### Act 2.3 - Reflexión Actividad Integral estructura de datos lineales

En esta actividad pude implementar otro algoritmo el cual fue Quicksort el cual tiene una complejidad promedio de  $O(n \log n)$ , el número total de comparaciones que

hace el algoritmo es:  $n + n + n + \dots + n = kn$

donde:  $k = \log_2(n)$ , por tanto el Orden de Complejidad del algoritmo en el mejor de los casos es  $O(n \cdot \log_2 n)$ . (colaboradores Wikipedia, 2021). También implementé el algoritmo de búsqueda Binaria que tiene una complejidad de  $O(\log n)$ , donde su expresión viene de  $T(n) = T(n/2) + k$ ,  $k$  is a constant. Este resultado de esta recurrencia da  $\log n$ , y la complejidad temporal es del orden de  $O(\log n)$ . Es más rápido que la búsqueda lineal y la búsqueda por salto. (Jindal, 2021)

También me dí cuenta sobre la importancia de poder usar diferentes estructuras de datos, en esta ocasión comparada con la anterior se usó una Doubly Linked List, una lista doblemente enlazada es una estructura de datos enlazados que consta de un conjunto de registros enlazados secuencialmente llamados nodos.

(colaboradores Wikipedia, 2021).

Y desde las primeras pruebas me di cuenta que de que era más eficiente que el otro, simplemente por el tiempo que tardaba en correr el programa pude darme cuenta.

Es un reto poder implementar y reajustar todas las funciones para que sean dobles usando nodos, dado que tuve que replantear todo el programa con respecto al pasado y que lo nuevo funcione al 100 con el programa que estamos haciendo.

Referencias:

colaboradores de Wikipedia. (2021, 4 mayo). Quicksort. Recuperado de <https://es.wikipedia.org/wiki/Quicksort#:~:text=El%20orden%20de%20complejidad%20del%20algoritmo%20es%20entonces%20de%20O,encuentran%20ordenadas%2C%20o%20casi%20ordenadas>.

Jindal, H. (2021, 11 marzo). Búsqueda binaria. Recuperado de [https://www.delftstack.com/es/tutorial/algorithm/binary-search/#:~:text=Complejidad%20del%20algoritmo%20de%20b%C3%BAsqueda%20binaria,-Complejidad%20del%20tiempo&text=Cuando%20realizamos%20la%20b%C3%BAsqueda%20binaria,a%20la%20mitad%20cada%20vez.&text=Este%20resultado%20de%20esta%20recurrencia,orden%20de%20O\(logn\)%20](https://www.delftstack.com/es/tutorial/algorithm/binary-search/#:~:text=Complejidad%20del%20algoritmo%20de%20b%C3%BAsqueda%20binaria,-Complejidad%20del%20tiempo&text=Cuando%20realizamos%20la%20b%C3%BAsqueda%20binaria,a%20la%20mitad%20cada%20vez.&text=Este%20resultado%20de%20esta%20recurrencia,orden%20de%20O(logn)%20).

Wikipedia contributors. (2021, 10 abril). Doubly linked list. Recuperado de [https://en.wikipedia.org/wiki/Doubly\\_linked\\_list](https://en.wikipedia.org/wiki/Doubly_linked_list)

Luis Enrique Lemus Martínez