Luis Enrique Gutierrez

A00837481

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

Profesor:

Dr. Eduardo Rodriguez Tello

Reflexión Actividad Integradora 2

Estructuras de Datos Lineales

Listas y Listas Doblemente Ligadas

Al indagar en el ámbito tecnológico la información puede ser guardada en un sinnúmero de tipos de algoritmos. En este caso estaremos hablando sobre estructuras de datos lineales, los cuales se basan en algún tipo de dato tenga un antecesor y un predecesor de manera sucesiva. El más común que se puede emplear en base a esta definición son las lista ligas ligas o también puede ser las lista doblemente ligadas.

Las listas lineales, como su nombre lo describe almacenan de manera lineal algún tipo de información por lo que algún tipo de dato está relacionado al siguiente y viceversa. La diferencia más grande existente entre este tipo de lista y las que son doblemente ligadas es que el segundo tipo se basan en tener dos nodos (derecho e izquierdo) los cuales se utilizan para acceder a datos anteriores y posteriores cada uno con su respectivo apuntador.

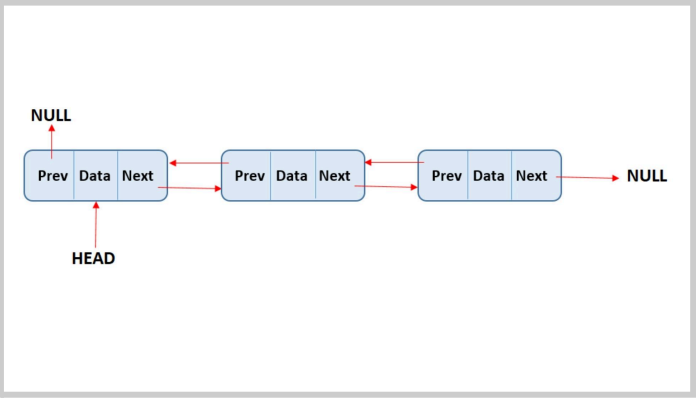

Imagen 1. Ejemplo de Listas doblemente ligadas

Obtenido de: https://www.boardinfinity.com/blog/a-detailed-walkthrough-of-doubly-linked-list/

En este caso la utilización de una lista doblemente ligada nos permite poder acceder a la información y movernos hacia delante y atrás de manera más eficiente. Lo anterior tomando en cuenta que la información a la que se le realizaron las diferentes operaciones es muy extensa por lo que utilizar una doubly linked list facilita el trabajo.

Desde una perspectiva similar podemos relacionar la efectividad de nuestro algoritmo en base a la complejidad temporal. Como fundamento (o por default) sabemos que tiene una complejidad O(1) al insertar o borrar algún elemento de la lista doblemente ligada. Lo anterior significa que solo se hizo una comparación para encontrar el elemento por lo tanto se realizó de manera rápida. En el caso particular de la actividad integradora esto cambia a una complejidad O(n) ya que tiene que atravesar la lista completa para realizar la operación. Muy

similarmente ocurre con los algoritmos de ordenamiento utilizado en el programa, para compararlos con la siguiente tabla.

Algoritmo	Complejidad
QuickSort	O(n log n)
MergeSort	O(n log n) // O(n)

Tabla 1: Comparación de complejidad temporal

Como resultado de ejecutar ambos algoritmos podemos llegar a una conclusión. Ambos son muy efectivos en el tema de ordenamiento teniendo en cuenta que tienen casi la misma complejidad dependiendo el caso. Al hacer un análisis de los resultados podemos ver que MergeSort es ligeramente más eficiente que QuickSort. Lo anterior se puede explicar de la siguiente manera, MergeSort cuando se ejecuta tiene complejidad de O(n) haciendo el trabajo más rápido que un algoritmo que presenta una complejidad de O(n log n) y asi se vio los resultados.

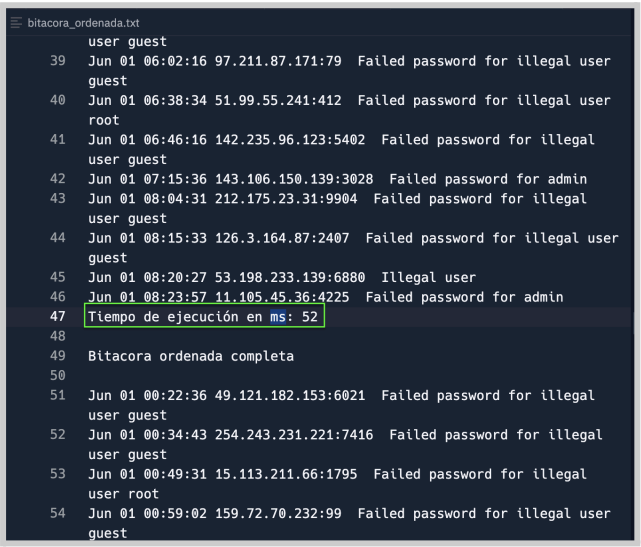

Imagen 2: Archivo de salida bitacora_ordenada.txt

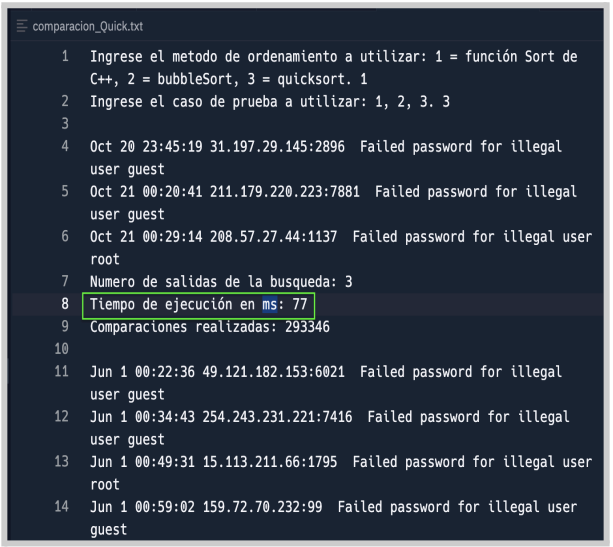

Imagen 3: Comparación Quick Sort

Se demostró ser un método eficiente en base a las complejidad temporal presentada y utilizada en el algoritmo. Por lo que la complejidad temporal juega un rol importante en la eficiencia de los algoritmos ejecutados. Finalmente, el utilizar listas doblemente ligadas es un aspecto poderoso para poder realizar diferentes tipos de operaciones, como lo puede ser insercion, borrar, u ordenar la información.

Referencias

GfG. (2024, February 8). *Time and space complexity of linked list*. GeeksforGeeks. https://www.geeksforgeeks.org/time-and-space-complexity-of-linked-list/

Henry, R. (2022, February 10). *¿Qué es una estructura de datos en programación?* https://blog.soyhenry.com/que-es-una-estructura-de-datos-en-programacion/#:~:text=Las%20estructuras%20de%20datos%20lineales,otro%20relacionados%20en%20forma%20lineal.

Listas Lineales. Linux, C/C++, Apuntes, etc... (2023, October 21). https://baulderasec.wordpress.com/programando-2/programacion-c-por-la-practica/capitulo-xv/listas-lineales/

Yadav, B. (2022, December 19). *Doubly linked list data structure in C++*. Scaler Topics. https://www.scaler.com/topics/doubly-linked-list-cpp/