

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey



Tecnológico de Monterrey

Programación de estructuras de datos y algoritmos fundamentales
Grupo 850

Después de clase | Tarea individual: Act-Integradora-2 Estructuras de Datos Lineales

Docente: Dr. Eduardo Arturo Rodríguez Tello

Víctor Huerta Loretz A01365532

20 de Abril de 2023

Utilizar diferentes estructuras de datos lineales, tiene un impacto positivo en el rendimiento final de un programa, en este caso utilizamos una doubly linked list para la resolución de este problema.

Una doubly linked list es una estructura de datos lineal en la cual los elementos son objetos separados con dos partes de referencia para cada una de los datos. En la cual la primera parte de referencia apunta al elemento anterior y la otra parte apunta al siguiente elemento en la lista. para así permitir una eficiente manera de eliminar y añadir elementos en la lista, mientras que una Linkedlist sólo apunta al siguiente elemento, dando así una eficiencia menor al eliminar y añadir elementos a la lista.

En temas de complejidad computacional podemos observar que en el caso de una doubly linked list las operaciones como inserción y eliminación son más eficientes ya que se puede acceder al nodo anterior, dándonos una complejidad de computacional de $O(1)$, por el fácil acceso al nodo anterior. Mientras que en una Linked List tiene una complejidad de $O(n)$ ya que es necesario recorrer la lista para encontrar el nodo anterior. Pero es importante considerar que en una doubly linked list ocupa más memoria por las partes de la referencia.

También se utilizó en este caso un merge sort el cual es un algoritmo de ordenamiento que divide el conjunto de datos en dos partes y vuelve a aplicar el algoritmo recursivamente a cada una de las partes. Se combinan ambas partes ordenadas para formar el conjunto de datos ordenado con una complejidad computacional de $O(n \log n)$.

Mientras que Quicksort tiene una complejidad de $O(n^2)$, en general quicksort suele ser más rápido con conjunto de datos más pequeños. Mientras que el mergesort es más eficiente en conjuntos de datos grandes que es el caso de esta actividad.

Por último se utilizo binary search para encontrar el rango de entradas dentro del registro, en la cual binary search cuenta con una complejidad computacional de $O(\log n)$ dándonos así una opción eficiente en un conjunto de datos grande como el que utilizamos en la actividad

Bibliografía:

- GeeksforGeeks. (s.f.). Quick Sort vs Merge Sort.
<https://www.geeksforgeeks.org/quick-sort-vs-merge-sort/>

- Stack Overflow. (s.f.). Quick Sort Vs Merge Sort.<https://stackoverflow.com/questions/680541/quick-sort-vs-merge-sort>
- GeeksforGeeks. (s.f.). Difference between Singly linked list and Doubly linked list.<https://www.geeksforgeeks.org/difference-between-singly-linked-list-and-doubly-linked-list/>
- Stack Overflow. (s.f.). When should I use a List vs a LinkedList.<https://stackoverflow.com/questions/169973/when-should-i-use-a-list-vs-a-linkedlist>
-