Act 2.3. Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales (Evidencia Competencia)

Alain Vicencio Arizabalo A01620758

Julian Andres Gonzalez Arguello A00831514

Programación de estructura de datos y algoritmos fundamentales

(Grupo 570)

ITESM

Viernes 27 de enero del 2022

Reflexión Individual Act 2.3

Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales

Reflexión Julián:

La primera entrega de la situación problema de la clase Programación de Estructura de Datos y Algoritmos Fundamentales trató sobre la creación de un programa que permitía la lectura de datos de una bitácora de texto, su organización utilizando algoritmos de ordenamiento, y finalmente su manejar con una búsqueda binaria para seleccionar datos específicos. El programa fue hecho con la finalidad de poder manipular un archivo para combatir intentos de hackeo e infiltración a una base de datos. La primera entrega almacenó los datos de la bitácora en un vector, mientras que el cambio en esta segunda entrega fue almacenar los datos en una Doubly Linked List.

Una Doubly Linked List es una estructura de datos lineal que contiene una serie de elementos con dos punteros, uno al elemento anterior y otro al elemento siguiente. Se diferencia de la Linked List ya que estas estructuras de datos lineales únicamente contienen un puntero al elemento siguiente. Las estructuras de datos lineales sirven para solucionar este tipo de problemas ya que su naturaleza permite el almacenamiento de datos y su manejo de forma eficiente. Para este problema se decidió utilizar una Doubly Linked List ya que su ligado doble permite el movimiento para recorrer y analizar los datos para enfrente como para atrás. Esto hace este tipo de estructuras más eficiente y conveniente para esta situación problema específico. Las operaciones básicas utilizadas en esta estructura son inserción, borrado, y búsqueda. La inserción es más eficiente en las Doubly Linked Lists y su complejidad es de O(1) al principio al igual que

al final de la lista. El borrado al igual que la búsqueda, tienen una complejidad de O(n), como en sus peores casos los métodos tuvieran que iterar sobre toda la lista para cumplir con su función.

Para ordenar la bitácora se utilizaron dos algoritmos de ordenamiento, siendo Quicksort y Mergesort. Quicksort tiene una complejidad temporal de O(n^2) en su peor caso y O(nLogn) en su caso promedio. Este algoritmo funciona al usar un pivote para dividir los elementos de mayores y menores, ordenando la lista recursivamente. Mergesort tiene una complejidad temporal de O(nLogn), y funciona al dividir la lista en sublistas, para después combinarlas y ordenarlas recursivamente. En nuestro programa, Mergesort fue más eficiente y se puede ver reflejado con su complejidad temporal, el algoritmo funciona más rápido y hace su función con un menor número de comparaciones.

Referencias

GeeksForGeeks. (2023). Difference between singly linked list and doubly linked list. Recuperado el 27 de enero del 2023 de: https://www.geeksforgeeks.org/difference-between-singly-linked-list-and-doubly-linked-list/

GeeksForGeeks. (2023). Introduction and insertion in a doubly linked list. Recuperado el 27 de enero del 2023 de: https://www.geeksforgeeks.org/difference-between-singly-linked-list-and-doubly-linked-list/