Instituto Tecnológico y de Estudios Superiores de Monterrey

TC1031.701 Programación de Estructura de Datos y Algoritmos Fundamentales

Act 2.3 - Actividad Integral estructura de datos lineales

Yahir Rivera Huerta A00572029
Saul Castañeda Carrillo A01541099
Luisa Castaños A01366643

Profesor: Eduardo Arturo Rodríguez Tello

Complejidad computacional y reflexiones finales

Fecha de Entrega: 24 de abril del 2022

Complejidad computacional

Algoritmo	Mejor	Promedio	Peor
leerArchivo	O(1)	O(n/2)	O(n)
nuevoArchivo	O(1)	O(n/2)	O(n)
busquedaFechas	O(log n)	O(1)	O(1)

Refelxiones Individuales

Yahir:

Durante la realización de esta actividad, junto con la actividad 1.3 pude entender un poco más sobre el lenguaje c++, ya que entendí un poco más sobre las funciones template, node, stack. Siento que estas funciones son realmente útiles y facilitan mucho a la hora de hacer programas multiusos y multifuncionales en cuestión a empresas.
Además de mencionar que el comprender la funcionalidad del iterative quick sort es de una gran ayuda, ya que son códigos que realmente se utilizan en el día a día y si no se acaban de entender bien de nada sirve aprenderlos. Creo que todo lo que estamos viendo en esta materia es de real ayuda para nuestro futuro profesional y quiero seguir aprendiendo más y más sobre las formas de programar.

Luisa:

No hay manera de describir la importancia y eficiencia del uso de los diferentes algoritmos de ordenamiento y búsqueda en una situación problema de la naturaleza ya que nos permiten buscar un elemento dentro de una colección y ordenar una colección en base a algún criterio, al igual de complementarse la una de la otra.

Son importantes los ordenamientos y búsquedas eficientes para optimizar el uso de otros algoritmos que requieren listas ordenadas para una ejecución rápida, también es es útil para poder datos en forma canónica o localizar un elemento con ciertas propiedades dentro de una estructura de datos y para generar resultados legibles

para cualquier persona y aunque fue difícil resolverlo eficientemente a pesar de un planteamiento simple y familiar, lo logramos satisfactoriamente

Esto no es solo para tener una calificación buena en las diferentes actividades pero también estamos aprendiendo habilidades para nuestro futuro profesional.

Saul:

Dado que en la programación existen distintos tipos de algoritmos de ordenamiento, se ha de utilizar el que en este caso nos genere una mayor eficacia en el programa. Por tal motivo decidimos utilizar un quickSort para el ordenamiento de nuestro archivo “Bitacora.txt”, el cual contiene datos de tipo Registro (clase), y al no ser una cantidad de datos exuberante. Optamos por usarlo debido a que, aunque el Merge sort realice menos comparaciones, requiere de un espacio de memoria adicional de O(n) el cual almacena la matriz adicional, mientras que en el quickSort necesita un espacio de O(registro n).

Y para este caso en específico, nos es mejor utilizar una Double Linked List, puesto que tenemos acceso al apuntar “prev”, el cual nos permite acceder, como su nombre lo dice, al nodo previo con el que estamos trabajando. Algo que la Linked List no nos proporciona. Por consiguiente, esto para el ordenamiento mediante un quickSort iterativo, nos fue muy conveniente, porque pudimos trabajar con el adaptador de contenedor denominado Stack, el cual le da al programador la funcionalidad de una pila, específicamente, una estructura de datos LIFO(Last In, First Out). Y así hacer el ordenamiento de nuestros datos de tipo Registro, mediante apuntadores.

Y dado que en las estructuras de datos lineales, se involucra un “single level inheritance”. El cual nos permite recorrer todos los elementos en una sola ejecución. Y a su vez son fáciles de implementar porque la memoria de la computadora está organizada de forma lineal, lo cual hace más eficiente el uso de este tipo de estructura de datos para este caso en específico.

Referencias

Iterative Quick Sort - GeeksforGeeks. (2022). Retrieved 23 April 2022, from https://www.geeksforgeeks.org/iterative-quick-sort/

std::stack - Español - Runebook.dev. Retrieved 24 April 2022, from https://runebook.dev/es/docs/cpp/container/stack

Diferencia entre Quick Sort y Merge Sort. (2019). Retrieved 24 April 2022, from https://es.gadget-info.com/difference-between-quick-sort

Difference between Linear and Non-linear Data Structures - GeeksforGeeks. (2022). Retrieved 24 April 2022, from https://www.geeksforgeeks.org/difference-between-linear-and-non-linear-data-structures/#:~:text=In%20linear%20data%20structure%2C%20single,queue%2C%20linked%20list%2C%20etc.