

Actividad 2.3

Actividad Integral estructura de datos lineales (Doubly Linked Lists)

Emma Fernanda Gómez Guerra (A00827097)

8 de octubre de 2020

Al igual que en la evidencia pasada, para esta evidencia se utilizaron los conocimientos sobre las metodologías y conceptos vistos en clase y consigo realizar un programa que finalmente se resolvió por medio de dos nodos uno tipo Doubly Linked List y otro para el "Queue.h". Como en la evidencia pasada, se le pedía al usuario datos para con ellos hacer la búsqueda de un determinado rango de información obtenidos del archivo sugerido por el proyecto: "bitacora.txt". Fue importante primero ordenar este archivo para que el programa pudiera leerlo de la manera más conveniente y se pudiera ejecutar con éxito.

Todo lo que ya se mencionó se hizo de manera casi idéntica a la de la Actividad 1.3, solo modificando algunos de los aspectos para hacer que pueda funcionar utilizando Doubly Linked Lists, Queue y Nodos para hacer su funcionamiento posible. Para entender mejor cómo funcionan las Doubly Linked List se utilizó la herramienta VisuAlgo. Y después de observar y entender su movimiento y realización de cada método (addLast, getData, updateAt, sort, merge, mergeSort, getSize, findData como búsqueda binaria, isEmpty y print) visto en clase, se creó un Nodo para la estructuración de estos datos de la Doubly Linked List con un "*prev*" y un "*next*" para indicar el valor anterior y el siguiente.

Recordando que con la búsqueda binaria los elementos de la lista pueden estar en cualquier orden. Y pueden llegar a hacerse n operaciones de comparación para poder obtener el resultado. Esta fue la búsqueda más eficiente para este tipo de problemas pues se puede hacer sobre un arreglo ordenado (como se hizo en la act 1.3 con el archivo 'bitacora.txt') y en esta actividad con el sort, merge y mergeSort. Su funcionamiento consiste en comparar si el valor buscado está en la mitad inicial o final. En la que esté, se divide en inicial y final de nuevo, y así hasta poder encontrar el valor que se está buscando.

Y la Doubly Linked List enlaza estos nodos (el nodo del Queue y el nodo de la Doubly Linked List) de manera secuencial. Un nodo consiste de tres partes: data, puntero al siguiente nodo en secuencia (next), puntero al nodo anterior (prev).

El método Sort se hizo con Queue, sabiendo que el Stack consiste de una estructura de datos linear en el cual los elementos pueden ser insertados y borrados únicamente de un extremo de la lista (top). Y el Queue serviría mejor para esta actividad pues el queue funciona de manera que el primero en entrar es el primero en salir. Utilizando el enqueue para insertar y el dequeue para borrar un elemento de la lista.

En conclusión, es importante saber manejar y entender cada uno de estos métodos para poder crear programas de manera fácil y eficiente. Para la actividad 2.3, entendí que su realización no era muy diferente a las de las actividades que ya habíamos completado y comentado en clase (Linked Lists, Queue y Stack) al igual que la actividad anterior. Solo se tuvieron que hacer pocas modificaciones para que se pudiera cumplir con los requisitos de esta actividad.