

**Instituto Tecnológico de Estudios Superiores de Monterrey**



**Tecnológico  
de Monterrey**

Campus Monterrey

**Act 2.3 - Actividad Integral estructura de datos lineales (Evidencia Competencia)**

**Programación de estructuras de datos y algoritmos fundamentales (Gpo 610)**

Eduardo López Benítez

**Fidel Morales Briones A01198630**

**Monterrey, Nuevo León, a 5 de octubre 2023**

**Reflexión:**

En este caso, se agregan los nodos a la lista ligada en un tiempo de  $O(1)$ , pero posteriormente se debe de utilizar un algoritmo que ordene los datos, se utilizó una versión iterativa de mergeSort() con una complejidad de tiempo de  $O(n \cdot \log(n))$  y una de espacio de  $O(1)$ . En un árbol de búsqueda binaria, la inserción de un nodo tiene una complejidad en el tiempo de  $O(\log(n))$ , pero la gran ventaja es que no es necesario ordenar ningún dato después de insertar todos los nodos, esto debido a que al momento de insertar un nodo, este se ordena automáticamente. Por lo que para los procesos combinados de inserción y ordenamiento de los nodos, un árbol de búsqueda binaria sí sería mejor.

Debido a que una lista ligada no tiene acceso directo a sus nodos con índices, no es posible utilizar el algoritmo de búsqueda binaria para realizar una búsqueda con un algoritmo mejor a  $O(n)$ . Sí es posible utilizar la búsqueda binaria, pero tendría la misma complejidad de tiempo que una búsqueda lineal. El árbol de búsqueda binaria tiene una complejidad de tiempo para búsquedas de  $O(\log(n))$ , por lo que también es mejor en este ámbito. Para la problemática de la evidencia, se pueden encontrar ambos valores del rango de fechas más cercanas o específicas introducidas por el usuario con una baja complejidad, y a partir de esas fechas se debería de recorrer el rango e imprimir los datos, esto ya tendría una complejidad en tiempo de  $O(n)$ .

La principal desventaja de un árbol de búsqueda binaria, es que cuando están desbalanceados, la eficiencia de estos disminuye mucho y se puede convertir en una estructura de datos lineal. Por lo que las búsquedas ya serían lineales y su principal atractivo de hacer una búsqueda binaria ya no estaría disponible. La otra desventaja sería que es difícil saber cuándo se tendrá un árbol balanceado, se debe de tener un conocimiento previo de los datos a procesar para saber si el árbol sufrirá un desbalance o no.

**Bibliografia:**

GeeksforGeeks. (2022). Iterative merge sort for linked list. *GeeksforGeeks*.

<https://www.geeksforgeeks.org/iterative-merge-sort-for-linked-list/>

GeeksforGeeks. (2023a). Introduction to Binary Search tree data structure and algorithm tutorials. *GeeksforGeeks*.

<https://www.geeksforgeeks.org/introduction-to-binary-search-tree-data-structure-and-algorithm-tutorials/>

GeeksforGeeks. (2023b). Applications Advantages and Disadvantages of Binary Search Tree. *GeeksforGeeks*.

<https://www.geeksforgeeks.org/applications-advantages-and-disadvantages-of-binary-search-tree/>