Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

Programación de estructuras de datos y algoritmos fundamentales (Gpo 573)

Profesores: Dr. Eduardo Arturo Rodríguez Tello

Andres Romo Castañeda A01234579

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

23/01/2026

Monterrey, Nuevo León

Reflexión Individual - Actividad 2.3

En esta actividad dimos un salto importante al dejar atrás los vectores y empezar a trabajar con listas doblemente ligadas. Al principio pensé que sería casi lo mismo, solo cambiando la forma de guardar datos, pero programando me di cuenta de que perder el acceso directo a las posiciones como solía hacerlo con con vectores hizo que esta actividad fuera un poco más difícil. Sinceramente, extrañé tener índices rápidos, porque aquí para llegar a un dato tienes que recorrer toda la lista nodo por nodo, y eso me hizo ver el costo real de la complejidad O(n) contra la O(1) que teníamos antes. Lo más interesante fue comparar mergesort contra quicksort en esta estructura. En la actividad pasada quicksort parecía muy bueno, pero aquí noté que el mergesort es el más eficiente para las listas ligadas. Entre nuestro equipo notamos que quicksort sufre más que mergesort porque tiene que estar recorriendo la lista para particionar o intercambiar valores, y mergesort fluye natural porque solo se trata de desconectar y reconectar punteros next y prev. No necesitas memoria extra ni mover datos pesados, solo mueves las flechas. Otra cosa que me llamó mucho la atención fue implementar la búsqueda binaria en una lista. Por teoría sabemos que es O(log n), pero al programar el algoritmo divide a la mitad pero llegar a esa mitad en una lista te cuesta tiempo porque tienes que ir nodo por nodo. Entonces, aunque la lógica es eficiente, la estructura de datos la frena. Esto me enseñó que no basta con elegir un buen algoritmo y tienes que elegir el algoritmo que encaje con tu estructura de datos. Me gustó mucho cómo trabajamos en equipo para resolver los errores de los punteros. Yo personalmente ya había trabajado con punteros pero no de este nivel. Verdaderamente es una herramienta potente pero compleja y fácil de corregir de que algo salga mal. Discutir entre nosotros por qué el programa tronaba o por qué no ordenaba bien nos ayudó a pulir la lógica. Al final quedamos bastante satisfechos de cómo quedó nuestro código.

Referencias:

- GeeksforGeeks. (2023, 14 de diciembre). *Merge Sort - Data Structure and Algorithms Tutorials*. https://www.geeksforgeeks.org/merge-sort/
- Programiz. (2024). *Binary Search Algorithm*. https://www.programiz.com/dsa/binary-search
- Simplilearn. (2023, 22 de noviembre). *Bubble Sort Algorithm: How it Works, Complexity, and More*. https://www.simplilearn.com/tutorials/data-structure-tutorial/bubble-sort-algorithm