

Actividad 2.3

Las listas ligadas son una estructura de datos simple que pueden servir para varias cosas, y tienen ventajas y desventajas muy claras.

Primero, las ventajas. Primero, este tipo de lista asignan memoria de forma dinámica a como van añadiendo datos, lo que hace que gasten muy poco espacio cuando no se necesitan, y pueden crecer y hacerse mas pequeñas en media ejecución del programa.

Otra cosa que cabe mencionar es que como se utilizan apuntadores para los nodos, se puede cruzar la lista en una sola dirección, lo que hace necesario el uso de apuntadores temporales, o recursión para cruzar toda la lista.

Todas las listas ligadas tienen una ventaja muy clara sobre otros tipos de estructuras de datos, que es la inserción en la cabeza, que siempre tiene una complejidad de $O(1)$, y en ciertas variantes que tienen el apuntador de la cola, la inserción en la cola también tiene una complejidad de $O(1)$. Aparte de esto, si se tiene acceso a un nodo, la inserción, edición y acceso de este son todos constantes.

También tiene un tiempo de lectura de la lista de $O(n)$, donde n es el número de nodos y es posible utilizar el método para ordenar Merge Sort sin necesidad de utilizar espacio extra, lo que nos deja tener un método de ordenamiento con unas de las mejores complejidades de tiempo y espacio, siendo $O(n \log n)$ y $O(1)$ respectivamente.

Esto hace que sea muy buena en situaciones en las que nada mas es necesario insertar datos y ordenarlos con muy poco espacio extra en el dispositivo, ya que no se necesita espacio extra en lo absoluto aparte de la propia lista, y en un tiempo absolutamente razonable.

Pero este tipo de lista también tiene una serie de desventajas bastante claras, por ejemplo, la búsqueda en este tipo de lista esta limitada a una complejidad de $O(n)$, ya que se necesitan visitar todos los nodos antes del nodo buscado para encontrarlo.

El problema principal yace en que no hay manera de acceder un dato que tome tiempo constante, lo que hace que el intentar búsqueda binaria en una lista ordenada implica que se tendría que pasar por la mitad de los nodos para llegar al nodo del medio para hacer la primera comparación, haciendo que la complejidad de tiempo mínima sea $O(n)$, la promedio $O(n)$ y la peor $O(n)$ para búsqueda binaria, y la búsqueda secuencial tiene una complejidad de $O(n)$ de por si en todas las estructuras de datos.

En general, podemos decir que para una actividad como la que estamos realizando, una lista doblemente encadenada no es particularmente ideal ya que queremos hacer operaciones de búsqueda en ella, que nos podemos dar cuenta no son muy rápidas en comparación con hasta un contenedor simple como un vector donde se puede hacer búsqueda binaria, que reduciría el tiempo de búsqueda una cantidad de tiempo muy exagerada.