Act 2.3 - Actividad Integral Estructura de Datos Lineales (Evidencia Competencia)

Alumno: Ricardo González Leal

Matrícula: A01639036

Materia: Programación de estructuras de datos y algoritmos fundamentales (TC1031)

Grupo: 13

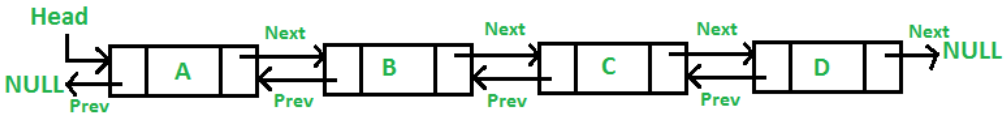
Profesor: Luis Ricardo Peña Llamas

Fecha: 15/10/21

Importancia y eficiencia del uso de las listas doblemente ligadas

Una lista doblemente ligada es una estructura de datos lineal, en la cual cada nodo contiene un puntero extra, comúnmente llamado “previous”, el cual tiene la función de apuntar al nodo anterior. Asimismo, se cuenta con el puntero “next” con el cual también cuentan las listas ligadas simples.

Dado que la lista doblemente enlazada permite el recorrido de nodos en ambas direcciones, se puede realizar un seguimiento del primer y último nodo. Siento este su principal diferenciador con respecto a las listas ligadas simples.



Aplicaciones importantes de las listas ligadas en situaciones problema similares:

- Las listas doblemente ligadas pueden ser utilizadas en sistemas de navegación en donde la navegación tanto del front end como del back sea requerida.
- Son utilizadas en los buscadores para implementar las funcionalidades de “ir hacia atrás” e “ir hacia adelante” al navegar por internet.
- Son utilizadas en múltiples programas y aplicaciones dentro de las funcionalidades de “Deshacer” y “Rehacer”.
- Otras estructuras de datos como stacks, tablas hash, árboles binarios también se pueden construir o programar usando una lista doblemente enlazada.
- La lista doblemente enlazada también se utiliza para construir la memoria caché MRU / LRU.
- En muchos sistemas operativos, el programador de subprocesos (lo que elige qué proceso debe ejecutarse en qué momento) mantiene una lista doblemente vinculada de todos los procesos que se ejecutan en ese momento.

Eficiencia de las listas ligadas:

Uso de memoria:

En comparación con los arreglos, las listas ligadas tienen un mejor uso de memoria. A diferencia de los arreglos, el tamaño de una lista ligada no está previamente definido; lo cual permite que la lista ligada pueda aumentar o disminuir de tamaño en el transcurso del programa. Y esto es posible gracias a que tanto para insertar y eliminar elementos de la lista ligada, los punteros se actualizan. Por lo cual, las operaciones de inserción y eliminar son muy rápidas, con un tiempo de complejidad de **O (1).**

Tiempo de inserción / eliminación:

Una de las desventajas de las listas ligadas tiene que ver con su tiempo de búsqueda, esto debido a que no se permite el acceso aleatorio, haciendo que no se se puede acceder a través de índices, a comparación de los arreglos. Por lo cual, las listas enlazadas requieren métodos iterativos que recorran todos los nodos hasta encontrar el nodo buscado. Esto ocasiona que su tiempo de complejidad sea de **O (n).**

Ventajas y Desventajas

Ventajas	Desventajas
Es de las estructuras de datos más fáciles de implementar.	Utiliza memoria adicional en comparación con los arreglos y la lista enlazada simple.
Permite el cruce de nodos en ambas direcciones	Dado que los elementos en la memoria se almacenan aleatoriamente, para acceder a un elemento es necesario recorrer la lista hasta encontrar el elemento.
La eliminación de nodos es fácil, gracias a los punteros con los que cuenta.	
Revertir la lista es simple y sencillo.	
Puede asignar o desasignar memoria fácilmente cuando sea necesario durante su ejecución.	
Es una de las estructuras de datos más eficientes para implementar cuando se requiere atravesar en ambas direcciones.	

Listas doblemente ligadas en esta situación problema

Considero que el hacer uso de una lista doblemente ligada para esta situación problema no es necesario. Esto principalmente debido a que para la actividad que se nos presenta, se cuenta con una bitácora la cual ya cuenta con un número fijo de registros el cual no va a cambiar, además, para las operaciones que se nos solicita se tienen que realizar múltiples búsquedas y ordenamientos que involucren los más de 16,000 registros con los que se cuenta, y bien como se vio en las desventajas de las listas ligadas, en ellas no se permite el acceso aleatorio, haciendo que no se pueda acceder a través de índices, a comparación de los arreglos. Por lo cual, si se utilizara una lista ligada se tendrían que realizar una gran cantidad de métodos iterativos que recorran todos los registros y los vayan ordenando según los requisitos que se piden, lo cual considero que sería muy ineficiente y complejo.

La manera en la que nosotros resolvimos este problema fue haciendo uso de vectores, con los cuales fácilmente pudimos realizar el ordenamiento ascendente por fecha utilizando el método de QuickSort. Posteriormente implementamos un Map (o Diccionario) en el cual se estuvo calculando y almacenando el número de IPs coincidentes dentro de un mismo día. Finalmente se ordenó el map para que esta información se mostrara ordenada ascendentemente por la cantidad de accesos por día.