



Act 2.3: Reflexión sobre Listas Doblemente ligadas

Instituto Tecnológico y de Estudios Superiores de Monterrey

Programación de estructuras de datos y algoritmos fundamentales

20/10/2021

Profesores:

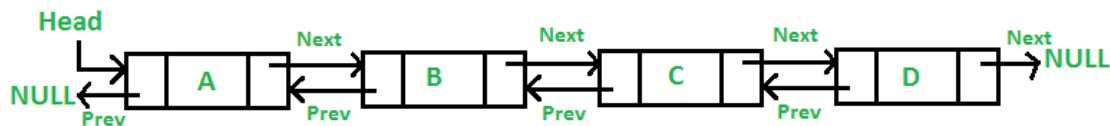
Luis Ricardo Peña Llamas

Elaborado por:

Pablo Agustín Ortega Kral A00344664

Reflexión sobre listas doblemente ligadas

Una lista doblemente ligada es una estructura de datos cuya unidad fundamental es el *nodo*, misma que contiene un valor y dos apuntadores: uno hacia el siguiente nodo en la lista y uno al previo. Esta estructura de datos es sumamente poderosa pues al ser lineal es fácil de programar e implementar y además cuenta con más versatilidad que una lista tradicional pues tiene apuntadores en dos direcciones.



En términos técnicos, las ventajas que ofrecen las listas doblemente ligadas provienen del apuntador extra con el que cuenta, siendo sus principales desventajas el uso de memoria que conlleva tener dos apuntadores en cada nodo. Algunas de sus peculiaridades son: que puede ser recorrida en ambos sentidos si se cuenta con el punto crítico *tail*, y que simplifican algunos procesos como la eliminación, siendo que el pointer al nodo anterior ya se conoce de antemano.

En cuanto a aplicaciones concretas, las listas doblemente ligadas puras tiene una variedad de utilidades, siendo algunas aplicaciones de alto nivel destacadas el permitir retroceder y avanzar entre dos objetos o estados: ya sea los botones de “atrás” y “adelante” de una página web o bien la funcionalidad de *undo* y *redo*. En cuanto a sus aplicaciones en el bajo nivel, las listas doblemente ligadas se utilizan en el *thread scheduler* para almacenar todos los procesos e irlos despachando apropiadamente.

Otra aplicación fundamental de las listas doblemente ligadas es fungir como la base de otras estructuras de datos, siendo que su abstracción de un nodo con datos y dos apuntadores puede aplicarse en una multitud de implementaciones. De esta manera, la estructura en cuestión sirve puede servir como base para construir pilas, colas, colas de prioridad, árboles binarios entre otros.

En el presente trabajo, se utilizó una cola de prioridad para realizar un insertion sort en base a la fecha del registro del error, sobre un vector previamente organizado por ocurrencia para presentar la información agrupada por fecha y en orden descendente de ocurrencia. Si bien el equipo eligió implementar dicha estructura con una lista ligada sencilla, también se podría haber usado una lista doblemente ligada, resultando con una complejidad de $O(N)$ en ambos casos.

Referencias:

Advantages, Disadvantages, and uses of Doubly Linked List - GeeksforGeeks. (2021).

Retrieved 21 October 2021, from

<https://www.geeksforgeeks.org/advantages-disadvantages-and-uses-of-doubly-linked-list/>

Pankaj, V. (2021). Data Structure : Doubly Linked List - Codeforwin. Retrieved 21 October

2021, from <https://codeforwin.org/2015/10/doubly-linked-list-data-structure-in-c.html>

Código

<https://github.com/PAOK-2001/ErrorSorter>