

Listas enlazadas

Nombre: Bernardo A. Márquez González

Materia: Estructura de datos

Profesor: Zaira Zubiria



Powered by Arizona State University

Fecha de entrega: 2023-06-22

Las listas ordenadas son una estructura de datos fundamental en programación. Permiten almacenar y organizar datos de manera secuencial, manteniendo un orden específico. En el lenguaje de programación C, es posible realizar operaciones de inserción en listas ordenadas, como pila (al principio), cola (al final) y en medio de la lista. Este resumen presenta una descripción de cada forma de inserción, su diagrama correspondiente y el código en C para implementar cada operación.

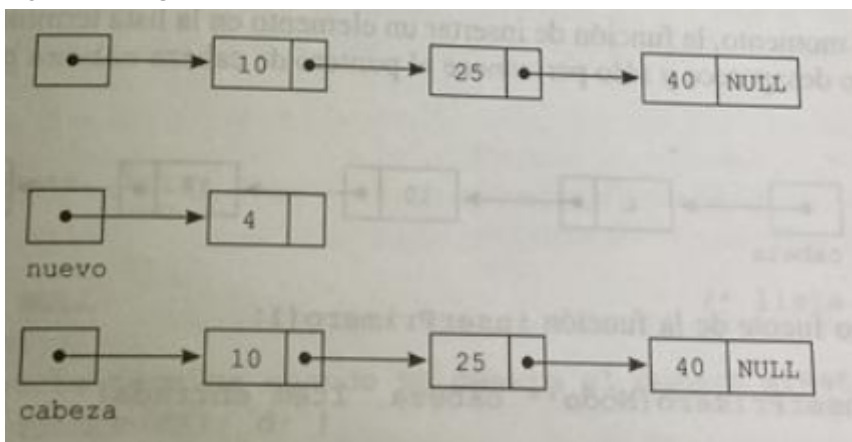
Inserción como pila

La inserción como pila implica agregar un elemento al principio de la lista. Es similar a apilar objetos, donde el último elemento apilado es el primero en ser desapilado. En una lista ordenada, se debe mantener el orden ascendente después de la inserción.

Código para realizar la inserción como pila:

```
void insertarPila(struct Nodo** cabeza, int dato) {  
    struct Nodo* nuevoNodo = (struct Nodo*)malloc(sizeof(struct  
Nodo));  
    nuevoNodo->dato = dato;  
    nuevoNodo->siguiente = *cabeza;  
    *cabeza = nuevoNodo;  
}
```

Ejemplo gráfico de la forma de inserción como pila:



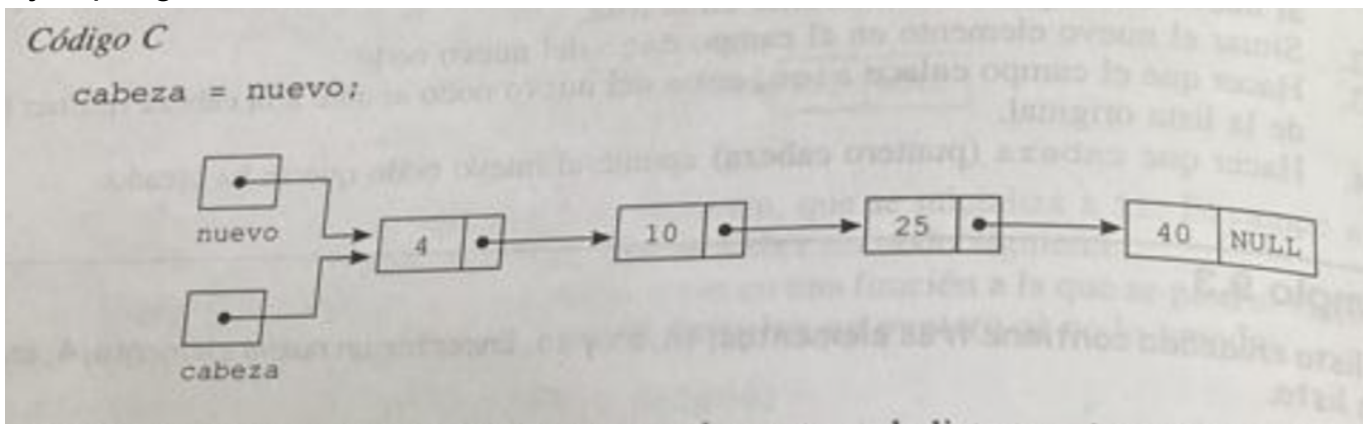
Inserción como cola

La inserción como cola implica agregar un elemento al final de la lista. Es similar a encolar objetos, donde el primer elemento encolado es el primero en ser desencolado. En una lista ordenada, se debe mantener el orden ascendente después de la inserción.

Código para realizar la inserción como cola:

```
void insertarCola(struct Nodo** cabeza, int dato) {  
    struct Nodo* nuevoNodo = (struct Nodo*)malloc(sizeof(struct  
Nodo));  
    nuevoNodo->dato = dato;  
    nuevoNodo->siguiente = NULL;  
  
    if (*cabeza == NULL) {  
        *cabeza = nuevoNodo;  
    } else {  
        struct Nodo* actual = *cabeza;  
        while (actual->siguiente != NULL) {  
            actual = actual->siguiente;  
        }  
        actual->siguiente = nuevoNodo;  
    }  
}
```

Ejemplo gráfico de la forma de inserción como cola:



Inserción en medio:

La inserción en medio implica agregar un elemento en una posición específica de la lista, manteniendo el orden ascendente. Se requiere encontrar el lugar adecuado para insertar el nuevo elemento.

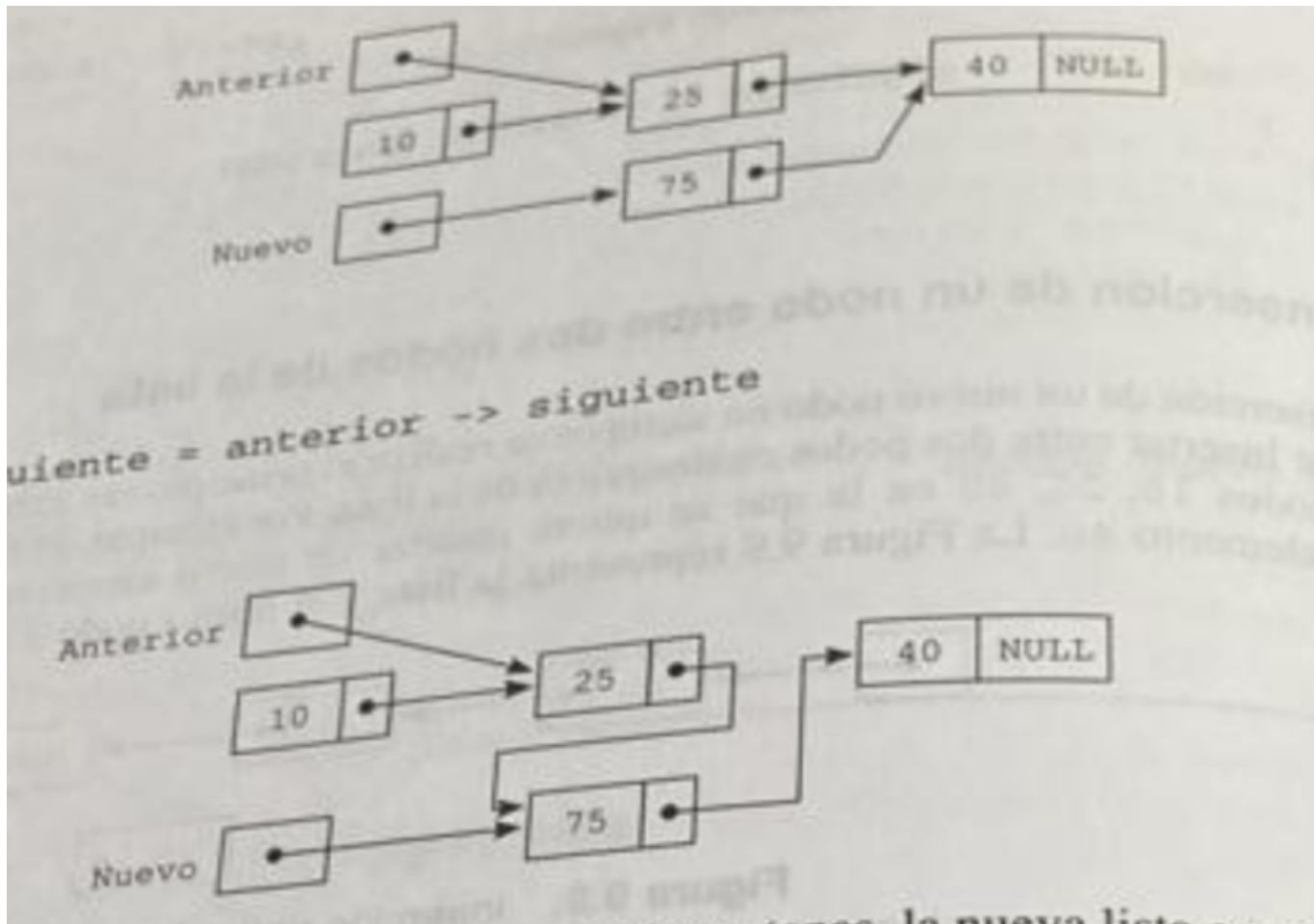
Código para realizar la inserción en medio:

```
void insertarMedio(struct Nodo** cabeza, int dato) {
    struct Nodo* nuevoNodo = (struct Nodo*)

malloc(sizeof(struct Nodo));
    nuevoNodo->dato = dato;

    if (*cabeza == NULL || dato < (*cabeza)->dato) {
        nuevoNodo->siguiente = *cabeza;
        *cabeza = nuevoNodo;
    } else {
        struct Nodo* actual = *cabeza;
        while (actual->siguiente != NULL && actual->siguiente->dato <
dato) {
            actual = actual->siguiente;
        }
        nuevoNodo->siguiente = actual->siguiente;
        actual->siguiente = nuevoNodo;
    }
}
```

Ejemplo gráfico de la forma de inserción en medio:



Referencias:

1. Programiz. (s.f.). Linked List Insertion in C. Recuperado de <https://www.programiz.com/dsa/linked-list-insertion>
2. GeeksforGeeks. (s.f.). Insertion in a sorted singly linked list. Recuperado de <https://www.geeksforgeeks.org/insertion-in-a-sorted-singly-linked-list/>
3. Tutorialspoint. (s.f.). C Library - <stdlib.h>. Recuperado de https://www.tutorialspoint.com/c_standard_library/stdlib_h.htm
4. Zahonero, I., & Joyanes, L. (2004). Algoritmos y estructuras de datos: una perspectiva en C (1ra ed.). McGraw-Hill.
5. OpenAI. (2021). GPT-3.5