PRESENTACIÓN

Centro de Enseñanza Técnica Industrial

Equipo: #7
Tema: Lista Circular de Doble Encabezado
Integrantes:
Ernesto Merín Núñez 22300912
Emmanuel Buenrostro Briseño 22300891
Nombre Corto de Integrantes: Merín | Emmanuel
Carrera: Desarrollo de Software
Grado y Grupo: 5I1
Materia: Estructura de Datos
Profesor: Karla Areli Isaac Rodrígez
Fecha: 10/10/2024

INVESTIGACIÓN

Lista Circular Doblemente Enlazada

Historia

Las listas circulares doblemente enlazadas son una evolución de las estructuras de datos enlazadas, desarrolladas para mejorar la eficiencia en ciertas operaciones de inserción y eliminación en listas lineales. Estas listas son una variante de las listas enlazadas que se remonta a los inicios del desarrollo de estructuras de datos avanzadas, como se explora en textos clásicos de algoritmos, como "Introduction to Algorithms" de Cormen et al. Estas listas han sido utilizadas desde los años 1960 para aplicaciones donde la memoria y la eficiencia de búsqueda son clave.

Definición

Una lista circular doblemente enlazada es una estructura de datos en la que cada nodo tiene dos punteros, uno al nodo anterior y otro al nodo siguiente, formando una estructura cíclica donde el último nodo apunta al primero y el primero apunta al último. A diferencia de las listas lineales, esta lista no tiene extremos "null".

Características

- **Circularidad:** El último nodo apunta al primero y viceversa, lo que permite una navegación continua.
- **Doble enlace:** Cada nodo tiene un puntero al nodo siguiente y al nodo anterior.
- **Sin nodos nulos:** Todos los punteros son válidos, ya que la lista es cíclica.
- **Acceso bidireccional:** Es posible navegar tanto hacia adelante como hacia atrás en la lista.

Composición

Cada nodo de una lista circular doblemente enlazada tiene tres partes principales:

- **Dato:** Contiene el valor o información del nodo.
- **Puntero anterior:** Apunta al nodo anterior en la lista.
- **Puntero siguiente:** Apunta al nodo siguiente en la lista.

Funcionamiento
El funcionamiento de una lista circular doblemente enlazada permite una navegación sin fin en ambas direcciones. Esta estructura es útil para aplicaciones que requieren operaciones frecuentes de inserción y eliminación en ambos extremos, como en simulaciones o en buffers circulares.

Estructura:

```
struct Nodo {
    int dato;
    Nodo* siguiente;
    Nodo* anterior;
};
```

Funciones Principales

- **Inserción:** La inserción en una lista circular doblemente enlazada puede realizarse al principio, en medio o al final de la lista. El proceso consiste en ajustar los punteros del nodo anterior y del nodo siguiente para incluir el nuevo nodo sin romper la circularidad.

- **Eliminación:** La eliminación en una lista circular implica remover un nodo ajustando los punteros de los nodos adyacentes para que apunten entre sí, eliminando así la referencia al nodo que se desea eliminar.

- **Búsqueda:** La búsqueda en esta lista implica recorrer la lista desde un nodo hasta que se encuentre el valor o se regrese al nodo de inicio.

Ventajas
- Navegación eficiente en ambas direcciones.
- Inserción y eliminación eficientes: No se requiere recorrer toda la lista para realizar operaciones en los extremos.
- Uso óptimo en aplicaciones cíclicas, como buffers circulares.

Desventajas
- Complejidad en la implementación: Más punteros que gestionar comparado con listas simplemente enlazadas.
- Mayor uso de memoria: Cada nodo tiene dos punteros adicionales en lugar de uno.

- Mayor tiempo de mantenimiento: La manipulación de los punteros es más costosa en términos de tiempo de programación y depuración.

Comparaciones

Comparación con Listas Circulares Simplemente Enlazadas:
Navegación: En una lista circular simplemente enlazada, cada nodo solo tiene un puntero hacia el siguiente nodo, lo que limita la navegación en una sola dirección. En cambio, las listas circulares doblemente enlazadas permiten el acceso en ambas direcciones gracias a los punteros dobles, lo que facilita operaciones como el retroceso y la eliminación desde cualquier punto.
Uso de memoria: Las listas simplemente enlazadas son más eficientes en términos de memoria, ya que solo requieren un puntero por nodo. Sin embargo, esto las hace menos flexibles para ciertas operaciones.
Aplicaciones: Las listas circulares simplemente enlazadas son útiles cuando solo se necesita recorrer la lista en una dirección, como en las colas circulares. Las listas doblemente enlazadas son preferibles cuando es necesario recorrer en ambas direcciones, como en simulaciones o en sistemas que requieren más flexibilidad.

Comparación con Listas Circulares Dobles con Encabezado:
Estructura del nodo: La principal diferencia es que las listas circulares doblemente enlazadas con encabezado incluyen un nodo adicional llamado "nodo cabeza", que no contiene datos, pero sirve como punto de referencia para facilitar las operaciones de inserción y eliminación. Este nodo actúa como un marcador que simplifica la gestión de la lista vacía o cuando solo hay un nodo.
Facilidad de uso: Las listas con nodo cabeza son más fáciles de manejar en términos de implementación, ya que el nodo cabeza permanece constante y evita los casos especiales para insertar o eliminar el primer o último nodo.
Aplicaciones: Las listas con encabezado son útiles en situaciones donde se realizan muchas operaciones en el principio o el final de la lista, mientras que las listas sin encabezado son preferibles cuando la eficiencia en el uso de memoria es crucial.

Ejemplos:

1. **Buffers Circulares en Sistemas Embebidos:**
   Un buffer circular es una estructura de datos eficiente usada en aplicaciones donde los datos se procesan en un ciclo continuo, como en sistemas embebidos que manejan entradas y salidas de datos de forma regular. Las listas circulares doblemente enlazadas permiten un acceso rápido en ambas direcciones, lo que facilita la inserción de nuevos datos y la eliminación de los más antiguos sin necesidad de reorganizar la estructura completa. En un sistema de procesamiento de datos en tiempo real, por ejemplo, una lista circular doblemente enlazada puede ayudar a mantener un flujo constante de datos en un orden cíclico.

2. **Simulaciones de Juegos de Cartas:**
   Las listas circulares doblemente enlazadas son útiles en simulaciones de juegos donde los jugadores participan de forma secuencial y cíclica. Un ejemplo claro son los juegos de cartas, donde un jugador toma una acción y luego el turno pasa al siguiente jugador. Esta estructura permite moverse hacia adelante o hacia atrás entre los jugadores, simulando un ciclo infinito en la participación.

3. **Sistemas Operativos:**
   En los sistemas operativos, se usan listas circulares doblemente enlazadas para la planificación de procesos. Los procesos se gestionan de manera circular, permitiendo que cada uno reciba una oportunidad de ejecutarse antes de que el turno regrese al primer proceso.

EXPOSICIÓN

Información Expuesta en Clase

Lista Circular Doblemente Enlazada

Historia
Estas listas son una variante de las listas enlazadas que se remonta a los inicios del desarrollo de estructuras de datos avanzadas, como se explora en textos clásicos de algoritmos, como "Introduction to Algorithms" de Cormen et al. Estas listas han sido utilizadas desde los años 1960 para aplicaciones donde la memoria y la eficiencia de búsqueda son clave.

Definición
Una lista circular doblemente enlazada es una estructura de datos en la que cada nodo tiene dos punteros, uno al nodo anterior y otro al nodo siguiente, formando una estructura cíclica donde el último nodo apunta al primero y el primero apunta al último.

Características
- Circularidad: El último nodo apunta al primero y viceversa, lo que permite una navegación continua.
- Doble enlace: Cada nodo tiene un puntero al nodo siguiente y al nodo anterior.
- Sin nodos nulos: Todos los punteros son válidos, ya que la lista es cíclica.
- Acceso bidireccional: Es posible navegar tanto hacia adelante como hacia atrás en la lista.

Composición
Cada nodo de una lista circular doblemente enlazada tiene tres partes principales:

Dato: Contiene el valor o información del nodo.
Puntero anterior: Apunta al nodo anterior en la lista.
Puntero siguiente: Apunta al nodo siguiente en la lista.

Funcionamiento
El funcionamiento de una lista circular doblemente enlazada permite una navegación sin fin en ambas direcciones. Esta estructura es útil para

aplicaciones que requieren operaciones frecuentes de inserción y eliminación en ambos extremos.

Inserción: Puede realizarse al principio, en medio o al final de la lista. El proceso consiste en ajustar los punteros del nodo anterior y del nodo siguiente para incluir el nuevo nodo sin romper la circularidad.

Eliminación: La eliminación en una lista circular implica remover un nodo ajustando los punteros de los nodos adyacentes para que apunten entre sí, eliminando así la referencia al nodo que se desea eliminar.

Búsqueda: La búsqueda en esta lista implica recorrer la lista desde un nodo hasta que se encuentre el valor o se regrese al nodo de inicio.

Estructura
Cada nodo contiene un valor entero (dato) y dos punteros: siguiente y anterior.

```
struct Nodo {
   int dato;
   Nodo* siguiente;
   Nodo* anterior;
};
```

Ventajas

- Navegación eficiente en ambas direcciones.
- La inserción y eliminación son eficientes, no se requiere recorrer toda la lista para realizar operaciones en los extremos.
- Uso óptimo en aplicaciones cíclicas, como buffers circulares.

Desventajas

- Complejidad en la implementación: Más punteros que gestionar comparado con listas simplemente enlazadas.
- Mayor uso de memoria: Cada nodo tiene dos punteros adicionales en lugar de uno.

Comparaciones

Comparación con Listas Circulares Simplemente Enlazadas:

Navegación:
En una lista circular simplemente enlazada, cada nodo solo tiene un puntero hacia el siguiente nodo, lo que limita la navegación en una sola dirección. En cambio, las listas circulares doblemente enlazadas permiten el acceso en ambas direcciones gracias a los punteros dobles, lo que facilita operaciones como el retroceso y la eliminación desde cualquier punto.
Uso de memoria:
Las listas simplemente enlazadas son más eficientes en términos de memoria, ya que solo requieren un puntero por nodo.

Aplicaciones:
Las listas circulares simplemente enlazadas son útiles cuando solo se necesita recorrer la lista en una dirección, como en las colas circulares. Las listas doblemente enlazadas son preferibles cuando es necesario recorrer en ambas direcciones, como en simulaciones o en sistemas que requieren más flexibilidad.

Comparación con Listas Circulares Dobles con Encabezado:

Estructura del nodo:
La principal diferencia es que las listas circulares doblemente enlazadas con encabezado incluyen un nodo adicional llamado "nodo cabeza", que no contiene datos, pero sirve como punto de referencia para facilitar las operaciones de inserción y eliminación. Este nodo actúa como un marcador que simplifica la gestión de la lista vacía o cuando solo hay un nodo.
Facilidad de uso:
Las listas con nodo cabeza son más fáciles de manejar en términos de implementación, ya que el nodo cabeza permanece constante y evita los casos especiales para insertar o eliminar el primer o último nodo.
Aplicaciones:
Las listas con encabezado son útiles en situaciones donde se realizan muchas operaciones en el principio o el final de la lista, mientras que las listas sin encabezado son preferibles cuando la eficiencia en el uso de memoria es crucial.

Ejemplos

1.Buffers Circulares en Sistemas Embebidos:
Un buffer circular es una estructura de datos eficiente usada en aplicaciones donde los datos se procesan en un ciclo continuo, como en sistemas embebidos que manejan entradas y salidas de datos de forma regular. Las listas circulares doblemente enlazadas permiten un acceso rápido en ambas direcciones, lo que facilita la inserción de nuevos datos y la eliminación de los más antiguos sin necesidad de reorganizar la estructura completa.

2. Simulaciones de Juegos de Cartas:
Las listas circulares doblemente enlazadas son útiles en simulaciones de juegos donde los jugadores participan de forma secuencial y cíclica. Un ejemplo claro son los juegos de cartas, donde un jugador toma una acción y luego el turno pasa al siguiente jugador. Esta estructura permite moverse hacia adelante o hacia atrás entre los jugadores, simulando un ciclo infinito en la participación.

3. Sistemas Operativos:
En los sistemas operativos, se usan listas circulares doblemente enlazadas para la planificación de procesos. Los procesos se gestionan de manera circular, permitiendo que cada uno reciba una oportunidad de ejecutarse antes de que el turno regrese al primer proceso.

Preguntas | Verdadero o Falso

1. En una lista circular doblemente enlazada, es posible recorrer los nodos tanto hacia adelante como hacia atrás. (Verdadero)

2. Las listas circulares simplemente enlazadas ocupan más memoria que las doblemente enlazadas. (Falso)

3. Las listas circulares doblemente enlazadas con encabezado facilitan las operaciones de inserción en una lista vacía. (Verdadero)

4. Los nodos en una lista circular simplemente enlazada tienen punteros tanto al siguiente como al nodo anterior. (Falso)

5. Una lista circular doblemente enlazada es útil en aplicaciones donde los datos deben ser procesados de manera cíclica. (Verdadero)

Referencias Bibliográficas:

- Javatpoint. (s. f.). Circular doubly linked list. Javatpoint. Recuperado el 3 de octubre de 2024, de https://www.javatpoint.com/circular-doubly-linked-list
- Simplilearn. (2022). What is a circular doubly linked list? Advantages and disadvantages. Simplilearn. Recuperado el 3 de octubre de 2024, de https://www.simplilearn.com/tutorials/data-structure-tutorial/circular-doubly-linked-list
- StudiousGuy. (s. f.). Circular doubly linked list in data structure. StudiousGuy. Recuperado el 3 de octubre de 2024, de https://www.studiousguy.com/circular-doubly-linked-list-in-data-structure/

Libros de Referencia:

- Tenenbaum, A. M., & Augenstein, M. J. (1996). Data structures using C. Prentice Hall.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms (3.ª ed.). MIT Press.