

Universidade Federal do Ceará - Campus Quixadá
QXD0010 – Estruturas de Dados
Prof. Atílio Gomes Luiz

1 Listas Circulares Duplamente Encadeadas

A estrutura de lista simplesmente encadeada, vista durante a aula, caracteriza-se por formar um encadeamento simples entre os nós: cada nó armazena um ponteiro para o próximo elemento da lista. Dessa forma, não temos como percorrer eficientemente os elementos em ordem inversa. O encadeamento simples também dificulta a retirada de um elemento da lista. Mesmo se tivermos o ponteiro do elemento que desejamos retirar, temos de percorrer a lista, elemento por elemento, para encontrar o elemento anterior, pois, dado o ponteiro para um determinado elemento, não temos como acessar diretamente seu elemento anterior.

Para solucionar esses problemas, podemos formar o que chamamos de **listas duplamente encadeadas**. Nelas, cada elemento tem um ponteiro para o próximo elemento e um ponteiro para o elemento anterior. Assim, dado um elemento, podemos acessar os dois elementos adjacentes: o próximo e o anterior. A lista duplamente encadeada pode ou não ter um nó sentinela e pode ou não ser circular, conforme as conveniências do programador. Uma **lista circular duplamente encadeada com Nó Sentinela** é uma lista duplamente encadeada na qual o último elemento da lista passa a ter como próximo o primeiro elemento, que, por sua vez, passa a ter o último como anterior. A Figura 1 ilustra uma lista duplamente encadeada com estrutura circular e a presença de um nó sentinela.

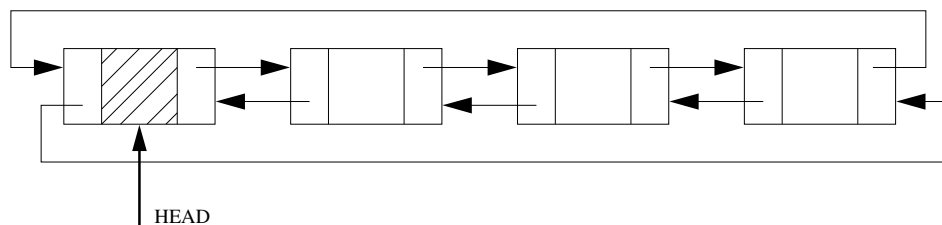


Figura 1: Lista circular duplamente encadeada com nó sentinela.

Problema: Implemente em C o Tipo Abstrato de Dados LISTA LINEAR usando como base a estrutura de dados LISTA CIRCULAR DUPLAMENTE ENCADEADA. A sua estrutura de dados deve suportar as seguintes operações:

- Criar uma lista vazia.
- Destruir a lista. Libera memória previamente alocada.
- Inserir um elemento ao final da lista.
- Inserir um elemento no início da lista.
- Remover um elemento do final da lista.
- Remover um elemento do início da lista.

- Retornar o tamanho da lista.
- Retornar se a lista está vazia.
- Imprimir os elementos da lista.
- Remover todos os elementos da lista e deixar apenas o nó sentinela. Ou seja, deixar a lista vazia.
- Remover um elemento que esteja num índice especificado. Lembre que numa lista de tamanho n os índices vão de 0 a $n - 1$.
- Inserir um elemento num índice especificado. Lembre que numa lista de tamanho n os índices vão de 0 a $n - 1$.
- Remover da lista todas as ocorrências de elemento com valor k ; esse k é passado para a função como argumento.
- Reverter a lista. Fazer em tempo $O(n)$ e sem usar memória adicional.
- Concatenar duas listas.
- Clonar uma lista. Um clone de uma lista é uma cópia da lista.
- Retornar se duas listas são iguais.
- Dadas duas listas ordenadas ℓ_1 e ℓ_2 , intercalar as duas listas gerando uma lista ℓ_3 ordenada. A lista ℓ_3 é formada pelos nós das listas ℓ_1 e ℓ_2 . Não aloque nós adicionais. Ao final desta operação, as listas ℓ_1 e ℓ_2 ficam vazias.
- Implementar o algoritmo BubbleSort para listas.
- Implementar o algoritmo InsertionSort para listas.
- Implementar o algoritmo SelectionSort para listas.