

Trabalho Prático: Lista Duplamente Encadeada  
Nome: Luis Felipe de Andrade Marques  
Matrícula: 557238  
Curso: Ciência da Computação - Estrutura de Dados

O Trabalho Dirigido 4 é sobre criar uma Lista Duplamente encadeada de números inteiros. Nele, você vai trabalhar com funções como inserir, apagar e acessar elementos da lista.

### ### Principais Dificuldades

Notação Big-O:

criarNo:

Alocar memória demanda um tempo constante, ou seja,  $O(1)$

Atribuição de valores tem tempo de  $O(1)$ ;

Tempo da função  $O(1)$ ;

criarLista:

Retornar NULL demanda  $O(1)$

Portanto, a complexidade de tempo da função criarLista é de  $O(1)$ .

insere:

Cria um novo nó.  $O(1)$

Verificações em geral.  $O(1)$

Percorrer a lista até a posição, no pior caso, demanda  $O(n)$

A inserção, no pior caso ocorre em  $O(n)$

Portanto a complexidade da função é  $O(n)$

procurar:

Verificações em geral.  $O(1)$

O loop que percorre a lista demanda  $O(n)$  no pior caso

A comparação demanda  $O(1)$

A complexidade será  $O(n)$  no pior caso

removerValor:

O if que verifica se a lista está vazia demanda tempo constante  $O(1)$

O loop que percorre todos os elementos para encontrar o valor, tem tempo de  $O(n)$ ;

A verificação se o item foi encontrado demanda tempo constante  $O(1)$ ;

Os ajustes dos ponteiros para remover no demandam  $O(1)$

A complexidade de tempo da função será, no pior caso,  $O(n)$  e no melhor caso  $O(1)$ ;

obterPosicao:

O if de verificação se a posição é válida demanda  $O(1)$ ;

A inicialização de variável demanda  $O(1)$

O loop que percorre a lista até a posição tem em seu pior caso  $O(n)$

Complexidade da função será  $O(n)$  no caso mais grave

imprimir:

O if que verifica se a lista está vazia demanda tempo  $O(1)$ ;

O loop de exibição de valores da lista tem tempo de  $O(n)$ ;

A impressão final tem tempo de  $O(1)$ ;

imprimirReverso:

O if que verifica se a lista está vazia demanda tempo  $O(1)$ ;

O loop que vai ate o final da lista demanda  $O(n)$   
O loop de exibição de valores da lista tem tempo de  $O(n)$ ;  
A complexidade total sera  $O(n)$ ;

tamanho:

Inicializacao de variavel demanda  $O(1)$

Loop pra percorrer todos os elementos e fazer a contagem demanda  $O(n)$

A verificacao demanda  $O(1)$

A complexidade da funcao sera  $O(n)$