

Lista 1 - Listas Encadeadas

Estrutura de dados 2025.1

Lista encadeada simples

Considere a seguinte definição de lista encadeada simples.

```
1 class No:
2     def __init__(self, chave: int, proximo):
3         self.chave = chave
4         self.proximo = proximo
5
6 class ListaEncSimples:
7     def __init__(self):
8         self.cabeca = None
9
```

Problema 1 Cite duas vantagens e duas desvantagens da lista encadeada dupla em relação ao array (lista sequencial estática).

Problema 2 Implemente as seguintes funcionalidades na lista encadeada simples:

- Inserir no início.
- Inserir no final.
- Remover do início.
- Remover do final.

Problema 3 Escreva o código de uma função **tamanho(lista)** que recebe como parâmetro uma lista encadeada simples e retorna a quantidade de elementos que ela tem. Qual a complexidade?

Problema 4 Escreva o código de uma função **busca(lista, alvo)** que recebe como parâmetro uma lista encadeada simples e um valor alvo. A função retorna verdade (True) se há algum nó da lista L que possua a chave igual ao alvo. Retorna falso (False) caso contrário. Qual complexidade?

Problema 5 Escreva o código de uma função **ordenada(lista)** que recebe como parâmetro uma lista encadeada simples. A função deve retornar True

se a lista está ordenada, False caso contrário. Uma lista deve estar ordenada de forma crescente, ou seja, do menor para o maior. Qual complexidade?

Problema 6 Escreva o código de uma função **repetido(lista)** que recebe como parâmetro uma lista encadeada simples. A função retorna True se há nós com chaves repetidas na lista L. Caso contrário, retorna False. Complexidade esperada: $O(n^2)$.

Problema 7 Escreva o código de uma função **remove_primeiro(lista, alvo)** que recebe como parâmetro uma lista encadeada simples e um inteiro alvo. A função remove a primeira ocorrência do elemento alvo na lista. Qual a complexidade?

Problema 8 Escreva o código de uma função **concatena(lista1, lista2)** que recebe como parâmetro duas listas encadeadas simples. A função concatena a lista2 no final da lista1. Qual complexidade?

Recursão

Problema 9 Escreva o código de uma função recursiva **tamanho_rec(lista)** que recebe como parâmetro uma lista encadeada e retorna o seu tamanho.

Problema 10 Escreva o código de uma função recursiva **soma_rec(lista)** que recebe como parâmetro uma lista encadeada simples de inteiros e retorna a soma de seus elementos.

Problema 11 Escreva o código de uma função recursiva **somapar_rec(lista)** que recebe como parâmetro uma lista encadeada simples de inteiros e retorna a soma de seus elementos.

Problema 12 Escreva o código de uma função recursiva **busca_rec(lista, alvo)** que recebe como parâmetro uma lista encadeada simples e um inteiro alvo. Retorna verdadeiro caso o alvo esteja dentro da lista.

Problema 13 Escreva o código de uma função recursiva **ordenada_rec(lista, alvo)** que recebe como parâmetro uma lista encadeada simples e retorna verdadeiro caso a lista esteja ordenada de forma crescente ou falso caso contrário.

Lista encadeada dupla

Considere a seguinte definição de lista encadeada dupla.

```

1 class No:
2     def __init__(self, chave: int, anterior, proximo):
3         self.chave = chave
4         self.anterior = anterior
5         self.proximo = proximo
6
7 class ListaEncDupla:
8     def __init__(self):
9         self.cabeca = None
10        self.cauda = None
11

```

Problema 14 Implemente as seguintes funcionalidades na lista encadeada dupla:

- Inserir no início.
- Inserir no final.
- Remover do início.
- Remover do final.
- Remove o K-ésimo elemento da lista.

Problema 15 Escreva o código de uma função **remove_primeiro(lista, alvo)** que recebe como parâmetro uma lista encadeada dupla e um inteiro alvo. A função remove a primeira ocorrência do elemento alvo na lista.

Problema 16 Escreva o código de uma função **converte_array_LED(array)** que recebe como parâmetro um array sequencial e o converte para uma lista encadeada dupla. A função remove a primeira ocorrência do elemento alvo na lista. Complexidade esperada: $O(n)$.

Problema 17 Escreva uma função que **remove_m_elementos(lista, m)** que remove os primeiros **m** elementos de uma lista encadeada dupla. Complexidade esperada: $O(m)$.