

Estrutura de Dados em Python

Prof. Nisston Moraes Tavares de Melo

Lista de Exercício #02

- Revisão

Estrutura de dados em Python

Dra. Ana Carolina Costa de Oliveira
prof2121@iesp.edu.br

LISTA SEQUENCIAL

É uma lista linear na qual a ordem lógica dos elementos (a ordem “vista” pelo usuário) é a mesma ordem física (em memória principal) dos elementos. Isto é, elementos vizinhos na lista estarão em posições vizinhas de memória.

LISTA SEQUENCIAL

- Controlar quais jogadores estão presentes no campo de treino (estrutura de dados)
 - Várias ações poderiam ser executadas
 - Inserir um jogador na estrutura de dados quando ele chegar ao campo
 - Verificar se um determinado jogador está presente, pesquisando o número do jogador na estrutura
 - Remover um jogador da estrutura de dados quando ele for para casa



LISTA SEQUENCIAL – INSERÇÃO

- Inserção de um novo jogador dentro do vetor

8	3	2	6	1		
---	---	---	---	---	--	--

- Único passo (inserido na primeira célula vaga do vetor)
- O algoritmo já conhece essa localização porque ele já sabe quantos itens já estão no vetor
- O novo item é simplesmente inserido no próximo espaço disponível
- Big-O constante – $O(1)$

LISTA SEQUENCIAL – PESQUISA LINEAR

- Percorrer cada posição do vetor
- Melhor caso: 8
- Pior caso: 1 ou número que não existe
- Em média, metade dos itens devem ser examinados ($N/2$)
- Big-O linear – $O(n)$

8	3	2	6	1		
---	---	---	---	---	--	--

LISTA SEQUENCIAL – EXCLUSÃO

8	3	2	6	1		
8	3	2	6	1		
8	3	6		1		
8	3	6	1			

LISTA SEQUENCIAL – EXCLUSÃO

- Pesquisar uma média de $N/2$ elementos (pesquisa linear)
 - Pior caso: N
- Mover os elementos restantes ($N/2$ passos)
 - Pior caso: N
- Big-O – $O(2n) = O(n)$

LISTA SEQUENCIAL – DUPLICATAS

- Deve-se decidir se itens com chaves duplicadas serão permitidos
- Exemplo de um arquivo de funcionários
 - Se a chave for o número de registro
 - Se a chave for o sobrenome
- Pesquisa: mesmo se encontrar o valor, o algoritmo terá que continuar procurando até a última célula (N passos)
- Inserção: verificar cada item antes de fazer uma inserção (N passos)
- Exclusão do primeiro item: $N/2$ comparações e $N/2$ movimentos
- Exclusão de mais itens: verificar N células e mais de $N/2$ células

LISTA SEQUENCIAL

```
import numpy as np
```

```
class Listasequencial:  
    def __init__(self, capacidade):  
        self.capacidade = capacidade  
        self.ultima_posicao = -1  
        self.valores = np.empty(self.capacidade, dtype=int)
```

LISTA SEQUENCIAL

```
# O(n)
def imprime(self):
    if self.ultima_posicao == -1:
        print('O vetor está vazio')
    else:
        for i in range(self.ultima_posicao + 1):
            print(i, ' - ', self.valores[i])
```

LISTA SEQUENCIAL

```
# O(1) - O(2)
def insere(self, valor):
    if self.ultima_posicao == self.capacidade - 1:
        print('Capacidade máxima atingida')
    else:
        self.ultima_posicao += 1
        self.valores[self.ultima_posicao] = valor
```

LISTA SEQUENCIAL

```
# O(n)
def pesquisar(self, valor):
    for i in range(self.ultima_posicao + 1):
        if valor == self.valores[i]:
            return i
    return -1
```

LISTA SEQUENCIAL

$O(n)$

```
def excluir(self, valor):  
    posicao = self.pesquisar(valor)  
    if posicao == -1:  
        return -1  
    else:  
        for i in range(posicao, self.ultima_posicao):  
            self.valores[i] = self.valores[i + 1]  
  
    self.ultima_posicao -= 1
```



Vamos
exercitar?

Exercício

1. Crie uma classe chamada Carro com um método acelerar() que imprime "O carro está acelerando!" e um método frear() que imprime "O carro está freando!".
2. Crie uma classe chamada Pessoa com um método cumprimentar(nome) que imprime "Olá, [nome]!".
3. Crie uma classe chamada Calculadora com métodos soma(a, b), subtracao(a, b), multiplicacao(a, b) e divisao(a, b) que realizam as operações correspondentes e retornam o resultado.
4. Crie uma classe chamada Retangulo com métodos calcular_area(largura, altura) e calcular_perimetro(largura, altura) que retornam a área e o perímetro de um retângulo, respectivamente.



Dúvidas?

Gabarito

```
class Carro:
    def acelerar(self):
        print("O carro está acelerando!")

    def frear(self):
        print("O carro está freando!")

meu_carro = Carro()
meu_carro.acelerar()
meu_carro.frear()
```

Gabarito

```
class Pessoa:
    def cumprimentar(self, nome):
        print(f"Olá, {nome}!")

pessoa = Pessoa()
pessoa.cumprimentar("Maria")
```

Gabarito

```
class Calculadora:
    def soma(self, a, b):
        return a + b

    def subtracao(self, a, b):
        return a - b

    def multiplicacao(self, a, b):
        return a * b

    def divisao(self, a, b):
        return a / b

calc = Calculadora()
print(calc.soma(10, 5))
print(calc.subtracao(15, 7))
print(calc.multiplicacao(4, 3))
print(calc.divisao(20, 4))
```

Gabarito

```
class Retangulo:
    def calcular_area(self, largura, altura):
        return largura * altura

    def calcular_perimetro(self, largura, altura):
        return 2 * (largura + altura)

retangulo = Retangulo()
print(retangulo.calcular_area(5, 3))
print(retangulo.calcular_perimetro(5, 3))
```



Dúvidas?

A dark blue, irregular ink blot or splash shape is centered on a white background. The blot has a textured, painterly appearance with some lighter blue and white speckles around its edges. Inside the blot, the text "Lista de Exercício." is written in a white, sans-serif font.

Lista de Exercício.



uniesp

Centro Universitário