|  |
| --- |
| **NOTAS DE AULA – PARADIGMAS DE PROGRAMAÇÃO** |

|  |  |
| --- | --- |
| **Curso:** Sistemas de Informação | **Ano:** 2023 |
| **Disciplina:** Paradigmas de Programação | **Semestre / Período:** 2º sem. / 2º período |
| **Docente Responsável:** Jean Marcelo Barbosa Cordeiro dos Santos | |
| **Aluno responsável: Victor Hugo Ferreira Bellini - 2023199588** | |

|  |  |
| --- | --- |
| Data | **02/08/2023** |
| Tema: | Introdução à disciplina |

**Foi apresentado o que será tratado nessa matéria**

|  |  |
| --- | --- |
| Data | **08/08/2023** |
| Tema: | Algoritmos de Pesquisa, Metodos de Orientação |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* **ORDENAR** algo é **SEMPRE** **IMPORTANTE** !
* **Pesquisa Binária:**

- Pré-ordem – **R-E-D**

- Em-ordem – **E-R-D**

- Pós-ordem – **E-D-R**

- Por-nível

* **Ordenação por troca**

- BubbleSort

* **Ordenação por seleção**

- SelectionSort

* **Ordenação por inserção**

- InsertionSort

* **Ordenação por interclasse**

- MergeSort

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**BubbleSort**

**Método da Bolha**

* **Características:**

Método simples.

Ordena com trocas sucessivas.

* **Como?**

Percorra o vetor comparando elementos adjacentes.

Troque posições se estiver fora de ordem.

Para um vetor de n elementos, **n – 1** varreduras são feitas para ordenar todos os elementos.

* **Exercicios:**

Caderno – Nº elementos 7

**{ca}derno a{cd}erno ac{de}rno acd{er}no acde{rn}o acden{ro} acdenor**

**{ac}denor a{cd}enor ac{de}nor acd{en}or acde{no}r acden{or} acdenor**

**{ac}denor a{cd}enor ac{de}nor acd{en}or acde{no}r acden{or} acdenor**

**{ac}denor a{cd}enor ac{de}nor acd{en}or acde{no}r acden{or} acdenor**

**{ac}denor a{cd}enor ac{de}nor acd{en}or acde{no}r acden{or} acdenor**

**{ac}denor a{cd}enor ac{de}nor acd{en}or acde{no}r acden{or} acdenor**

* **Eficiencia**

Custo alto de desempenho

Adequado para registros pequenos.

Inadequado para registros grandes.

* **Como otimizar?**

Finalizar execução quando nenhuma troca for realizada.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SelectionSort**

**Ordenação por Seleção**

* **Caracteristicas**

Metodo simples.

Ordena por seleções sucessivas.

* **Como?**

Realiza uma busca pelo elemento de menor valor.

Trocar com o elemento da primeira posição.

Realizar as rotinas com **n-1** elementos restantes, em seguida com os **n-2** elementos restantes, até que reste apenas **1** elemento.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**InsertionSort**

**Ordenação por Inserção**

* **Caracteristicas**

Metodo simples.

Ordena por inserções em vetor ordenado

* **Como?**

Considera 2 containers de elementos:

**Ordenado**

**Não ordenado**

Inserir um elemento por vez do container não-ordenado para o container ordenado

**Observação:**

O InsertionSort, funciona utilizando dois containers, um ordenado e um não-ordenado. O não ordenado irá ter n elementos e iremos mover esses elementos para o container ordenado. Após mover o primeiro elemento, iremos assimilar com o próximo elemento e caso se o elemento da direita for menor que o elemento da esquerda, iremos realocar para o lado esquerdo, caso contrário não iremos mudar de lado.

**MergeSort**

**Ordenação por Junção**

* **Caracteristicas**

Ordena por junções sucessivas

* **Como?**

Junções sucessivas de 2 containers ordenados em um único container ordenado

Divide o container em 2 regiões de comprimento (n/2)

Ordena recursivamente cada região

Faz a junção(merge) das regiões ordenadas

**Codigo:**

Com um container de **x** elementos:

Se x < 2

Retorna

Senão

Ordenar a metade do lado esquerdo

Ordenar a metade do lado direito

Juntar as duas metades

* **Observação**

No MergeSort, iremos ter um nº de elementos para resolvermos, e para efetuar essa situação, iremos dividi-la ao máximo por 2.

Exemplo: Se tivermos 8 números, divide por 2 e fica 4 de cada lado, dividindo novamente, 2 por 2 e novamente 1 por 1.

Após isso iremos relacionar e juntar caso necessário.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Data | **15/08/2023** |
| Tema: | Introdução ao JavaScript |

**Variaveis e tipo de dados**

Var nome\_variavel = valor\_inicial;

Ou

Var nome\_variavel;

**Tipos de Dados**

* Números inteiros
* Números de virgula flutuante
* Booleanos (verdade ou falso)
* Strings (conjunto de caracteres)
* Objetos ou funções
* Null (vazio ou nulo)

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

|  |  |
| --- | --- |
| Data | **22/08/2023** |
| Tema: | Segunda aula sobre o JavaScript |

**Imagens dos códigos feito em sala:**

Texto

Descrição gerada automaticamente

Uma imagem contendo Tabela

Descrição gerada automaticamenteTexto

Descrição gerada automaticamente

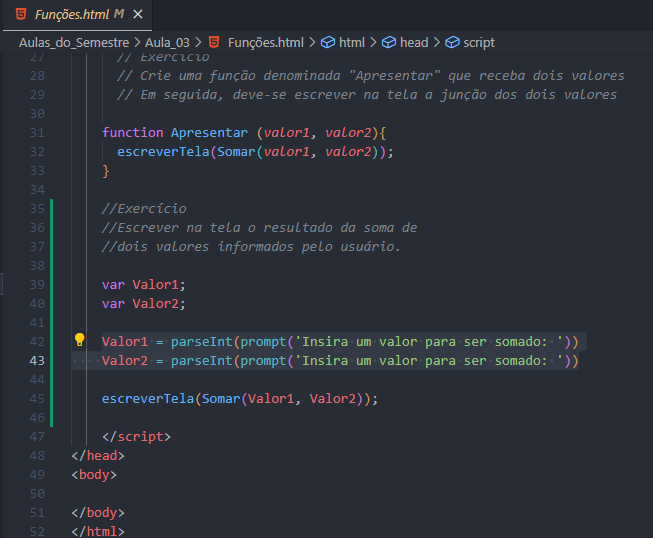
Texto

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

|  |  |
| --- | --- |
| Data | **29/08/2023** |
| Tema: | Terceira aula de JavaScript |



Texto

Descrição gerada automaticamente

Interface gráfica do usuário

Descrição gerada automaticamente com confiança baixa