

PANC: Projeto e Análise de Algoritmos

Projeto 02: Ordenações de Arrays

Breno Lisi Romano

<http://sites.google.com/site/blromano>

**Instituto Federal de São Paulo – IFSP São João da Boa Vista
Bacharelado em Ciência da Computação – 3º Semestre**



**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus São João da Boa Vista**



Problema: Ordenação (1)

- **Problema de Ordenação:**

- Ordenar uma sequência de números de maneira não decrescente

- **Entrada:**

- Uma sequência de n números $\langle a_1, a_2, a_3, \dots, a_n \rangle$

- **Saída:**

- Uma permutação $\langle a'_1, a'_2, a'_3, \dots, a'_n \rangle$ da sequência de entrada, tal que $a'_1 \leq a'_2 \leq a'_3 \leq \dots \leq a'_n$



Problema: Ordenação (2)

- Desenvolver uma **solução**, na **linguagem C**, em um mesmo projeto, com todos os algoritmos de ordenação que trabalhamos em aula (Inserção, Troca e Seleção). Os algoritmos são:
 - Métodos de Ordenação por Inserção
 - Direta
 - Binária
 - Shell Sort
 - Métodos de Ordenação por Troca
 - Bubble Sort
 - Shake Sort
 - Comb Sort
 - Quick Sort
 - Métodos de Ordenação por Seleção
 - Select Sort
 - Heap Sort
 - Outros Métodos de Ordenação
 - Merge Sort
 - Radix Sort



Problema: Ordenação (2)

- **Deve-se também propor um novo algoritmo de ordenação. Você precisa definir a lógica deste novo algoritmo e apresentar uma ilustração, com um exemplo prático, em um arquivo .pdf a ser entregue junto com o projeto. Apresente um estudo da complexidade do algoritmo proposto.**
 - Pode ser um algoritmo com base em qualquer outro que você aprendeu, desde que apresenta algo “novo”
 - Não se preocupem em propor um ótimo algoritmo, só pensem em criar uma proposta “diferente”



Orientações Gerais

- Para a **resolução do problema**:
 - O tamanho do array deve ser fornecido pelo usuário;
 - Os **elementos** do **array** deve ser **gerado aleatoriamente**, com apenas números **inteiros**, com base no tamanho **fornecido**
 - **Apresentar o resultado, passo a passo, da ordenação de cada um dos algoritmos de ordenação**
 - Deve-se imprimir um **resumo da quantidade de comparações e trocas** realizadas por cada um dos algoritmos de ordenação
 - Criar **a solução com a estrutura** de um projeto.
- Para **execução**:
 - O **Projeto 02** pode ser desenvolvido **em triplas (no máximo)**
 - **Data de Entrega**: 26/06/2023
 - **Local da Entrega**: Atividade específica no turma do Google Classroom



Exemplo Parcial da Solução

Algoritmos de Ordenacoes:

Entre com o tamanho (n) do Array: 10

Array Original Gerado Aleatoriamente:

49 60 39 8 7 93 45 60 19 84

Ordenando o Array - Insertion Sort():

49 60 39 8 7 93 45 60 19 84
7 8 19 39 45 49 60 60 84 93

Ordenando o Array - Insertion Sort Alternativo:

49 60 39 8 7 93 45 60 19 84
7 8 19 39 45 49 60 60 84 93

Ordenando o Array - MergeSort sem Sentinela:

49 60 39 8 7 93 45 60 19 84
7 8 19 39 45 49 60 60 84 93

Ordenando o Array - Ordenacao Binaria:

49 60 39 8 7 93 45 60 19 84
7 8 19 39 45 49 60 60 84 93

Analise dos Algoritmos de Ordenacao

	Comparacoes	Trocas
IS	45	20
MS	9	34
OB	22	20