

	CURSO: TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS	
	DISCIPLINA: ESTRUTURA DE DADOS II	CARGA HORÁRIA: 60
	COORDENADOR: JURANDIR CAVALCANTE LACERDA JÚNIOR	MÓDULO: III
	PROFESSOR: FELIPE GONÇALVES DOS SANTOS	ANO: 2024.1
ALUNO(A): ANA VITÓRIA DIAS CARVALHO		

ANÁLISE COMPARATIVA DOS ALGORITMOS DE ORDENAÇÃO

1. INTRODUÇÃO:

Os algoritmos de ordenação desempenham um papel fundamental na computação, sendo essenciais para organizar dados de maneira eficiente. Este trabalho tem como objetivo analisar e comparar o desempenho dos principais algoritmos de ordenação em três cenários distintos: vetores com um milhão de números em ordem crescente, em ordem decrescente e em ordem aleatória. Serão considerados critérios como tempo de execução, quantidade de comparações e quantidade de trocas realizadas pelos algoritmos. Através dessas análises, será possível avaliar a eficiência e a aplicabilidade de cada algoritmo em diferentes contextos.

2. DESCRIÇÃO DO EQUIPAMENTO UTILIZADO:

É relevante considerar o software e o hardware utilizado para realizar os testes, pois as características do sistema podem impactar diretamente nos resultados obtidos. O notebook onde foram realizados os testes opera com o sistema operacional Windows 10 Home Single Language, processador AMD Ryzen 3 3200U, 8GB de memória RAM DDR4 e tipo de armazenamento HDD de 1000GB. Diante das especificações apresentadas, é possível concluir que o processador e a memória RAM oferecem uma resposta razoável diante dos problemas propostos, no entanto, o armazenamento do tipo HDD pode limitar a velocidade de acesso aos dados em comparação com um SSD, especialmente em operações que envolvem leitura e escrita intensivas como as realizadas pelos algoritmos de ordenação utilizados.

3. RESULTADOS:

Nos testes realizados, seis algoritmos de ordenação foram analisados: *Bubble Sort*, *Insertion Sort*, *Selection Sort*, *Merge Sort*, *Quick Sort* e *Heap Sort*. Cada algoritmo foi aplicado para ordenar vetores com um milhão de números, provenientes de arquivos .txt em três cenários distintos, um milhão de números em ordem crescente, decrescente e em ordem aleatória, considerando como principais métricas de desempenho o tempo de execução e as quantidades de comparações e trocas realizadas. A seguir, são apresentados os resultados obtidos.

3.1. 1º CENÁRIO: VETOR COM UM MILHÃO DE NÚMEROS EM ORDEM CRESCENTE.

TABELA COMPARATIVA DOS ALGORITMOS DE ORDENAÇÃO			
1º CENÁRIO			
ALGORITMO	TEMPO DE EXECUÇÃO (EM SEGUNDOS)	COMPARAÇÕES	TROCAS
Bubble Sort	704	728.379.968	0
Insertion Sort	0,026	999.999	1.999.998
Selection Sort	366	1.783.293.664	999.999
Merge Sort	614	10.066.432	19.951.424
Quick Sort	0,085	20.951.423	0
Heap Sort	0,138	40.577.334	19.788.667

3.2. 2º CENÁRIO: VETOR COM UM MILHÃO DE NÚMEROS EM ORDEM DECRESCENTE.

TABELA COMPARATIVA DOS ALGORITMOS DE ORDENAÇÃO			
2º CENÁRIO			
ALGORITMO	TEMPO DE EXECUÇÃO (EM SEGUNDOS)	COMPARAÇÕES	TROCAS
Bubble Sort	1554	728.379.968	1.783.293.664
Insertion Sort	1.639	999.999	1.785.293.662
Selection Sort	530	1.783.293.664	1.499.999
Merge Sort	596	9.884.992	19.951.424
Quick Sort	0,086	20.951.424	500.000
Heap Sort	0,256	37.666.816	18.333.408

3.3. 3º CENÁRIO: VETOR COM UM MILHÃO DE NÚMEROS EM ORDEM ALEATÓRIA.

TABELA COMPARATIVA DOS ALGORITMOS DE ORDENAÇÃO			
3º CENÁRIO			
ALGORITMO	TEMPO DE EXECUÇÃO (EM SEGUNDOS)	COMPARAÇÕES	TROCAS
Bubble Sort	2.155	728.379.968	1.121.222.909
Insertion Sort	754	999.999	1.123.222.907
Selection Sort	261	1.783.293.664	1.999.982
Merge Sort	606	18.673.131	19.951.424
Quick Sort	0,197	28.877.516	4.127.523
Heap Sort	0,249	39.095.126	19.047.563