



Relatório do Trabalho AV

Algoritmos e Complexidade – 2024.1

Fernando José Tozato de Siqueira

202202760811

Introdução

Algoritmos de ordenação são sequências de comandos que ordenam uma lista de dados em alguma ordem. Para este trabalho, escolhi os algoritmos InsertionSort, de complexidade $O(n^2)$, e MergeSort, de complexidade $O(n \log n)$. Quando estava fazendo os testes, deixei dois programas principais rodando em computadores diferentes, em um deles usei multithreading, enquanto no outro não. Link do github: https://github.com/Fernando-Tozato/Trabalho_Algoritmos.

Resultados

O algoritmo com multithreading foi executado em um computador com um processador Xeon E3-1230 e 10Gb de memória ram, enquanto o algoritmo sem multithreading foi executado em um Intel i5 -8500XT com 12Gb de memória ram. Os resultados estão nas seguintes tabelas:

Com multithreading:

Tamanho	Algoritmo	Tempo
1000	InsertionSort	00:00:00.12
1000	MergeSort	00:00:00.00
10000	InsertionSort	00:00:10.11
10000	MergeSort	00:00:00.00
100000	InsertionSort	00:16:47.30
100000	MergeSort	00:00:00.00
1000000	InsertionSort	29:18:00.20
1000000	MergeSort	00:00:00.02



Estácio

Relatório do Trabalho AV

Algoritmos e Complexidade – 2024.1

Sem multithreading:

Tamanho	Algoritmo	Tempo
1000	InsertionSort	00:00:00.02
1000	MergeSort	00:00:00.00
10000	InsertionSort	00:00:02.76
10000	MergeSort	00:00:00.03
100000	InsertionSort	00:05:05.90
100000	MergeSort	00:00:00.32
1000000	InsertionSort	12:44:31.14
1000000	MergeSort	00:00:04.18

Conclusão

Achei surpreendente que mesmo usando a divisão de processos, o algoritmo mais complexo demorou muito mais tempo, enquanto o algoritmo menos complexo foi seu completo oposto. Talvez isso aconteça pela configuração das máquinas utilizadas, mas mesmo assim não era o resultado que eu esperava.