

Ficha de comparação de algoritmos de ordenação 2023/1

Nome do(a) aluno(a):	Eduarda Arianne Marques de Matos
----------------------	----------------------------------

Especificação do equipamento

Sistema Operacional	Windows 11
Processador	Intel Core i7 10th Geração
Arquitetura (32 ou 64 bits)	64 bits
Memória RAM Total	16 GB
Versão do Python	3.11.2

Para que a análise de eficiência (abaixo) tenha resultados comparáveis, todos os testes precisam ser realizados no mesmo equipamento (descrito acima).

Análise de eficiência

Anotar, no quadro abaixo, o tempo e a memória dispendidos para a execução de cada algoritmo trabalhando sobre diferentes tamanhos de amostra.

	Tamanho da amostra							
Algoritmo	10.000		25.000		50.000		100.000	
Bubble Sort	Tempo	3,46 minutos	Tempo	21,35 minutos	Tempo	86,17 minutos	Tempo	347,76 minutos
	Memória	0,00031 MB	Memória	0,00031 MB	Memória	0,00032 MB	Memória	0.00028228759765625 MB
Selection Sort	Tempo	1,33 minutos	Tempo	8,51 minutos	Tempo	25,5 minutos	Tempo	101,4 minutos
	Memória	0,0000392913818359375 MB	Memória	0,000392913818359375 MB	Memória	0,000396728515625 MB	Memória	0,000396728515625 MB
Merge Sort RECURSIVO	Tempo	0,29 segundos	Tempo	0,81 segundos	Tempo	1,77 segundos	Tempo	3,85 segundos
	Memória	0,23529052734375 MB	Memória	0,5919647216796875 MB	Memória	1,18841552734375 MB	Memória	2,29156494140625 MB
Merge Sort ITERATIVO	Tempo	465,33 milissegundos	Tempo	1345,44 milissegundos	Tempo	2902,01 milissegundos	Tempo	6490,38 milissegundos
	Memória	0,08346939086914062 MB	Memória	0,22388076782226562 MB	Memória	0,4474830627441406 MB	Memória	0,8946876525878906 MB
Quick Sort RECURSIVO	Tempo	379,51 milissegundos	Tempo	963,64 milissegundos	Tempo	2256,91 milissegundos	Tempo	5150,76 milissegundos
	Memória	0,00372314453125 MB	Memória	0,00555419921875 MB	Memória	0,01690673828125 MB	Memória	0,017364501953125 MB
Quick Sort ITERATIVO	Tempo	485,8 milissegundos	Tempo	1260,28 milissegundos	Tempo	2953,91 milissegundos	Tempo	6480,46 milissegundos
	Memória	0,07763671875 MB	Memória	0,19219970703125 MB	Memória	0,3829345703125 MB	Memória	0,7646484375 MB
MELHOR RESULTADO	Tempo	Quick Sort Recursivo (379,51ms)	Tempo	Quick Sort Recursivo (963,64ms)	Tempo	Quick Sort Recursivo (2256,91ms)	Tempo	Quick Sort Recursivo (5150,76ms)
	Memória	Bubble Sort (0,00031 MB)	Memória	Bubble Sort (0,00031 MB)	Memória	Bubble Sort (0,00032 MB)	Memória	Bubble Sort (0.00028228759765625 MB)

Passadas, comparações e trocas

Anotar, no quadro abaixo, o tempo e a memória dispendidos para a execução de cada algoritmo trabalhando sobre diferentes tamanhos de amostra.

	Tamanho da amostra							
Algoritmo	10.000		25.000		50.000		100.000	
Bubble Sort	Passadas	9.942	Passadas	24.675	Passadas	49.353	Passadas	99.259
	Comparações	99.410.058	Comparações	616.850.325	Comparações	2.467.600.647	Comparações	9.925.800.741
	Trocas	25.297.553	Trocas	157.485.392	Trocas	625.828.205	Trocas	2.498.177.562
Selection Sort	Passadas	9.999	Passadas	24.999	Passadas	49.999	Passadas	99.999
	Comparações	49.995.000	Comparações	312.487.500	Comparações	1.249.975.000	Comparações	4.999.950.000
	Trocas	9.990	Trocas	24.984	Trocas	49.991	Trocas	99.982
Quick Sort RECURSIVO	Passadas	6.739	Passadas	16.815	Passadas	33.606	Passadas	67.259
	Comparações	247.491	Comparações	616.240	Comparações	1.435.726	Comparações	3.148.139
	Trocas	74.724	Trocas	183.025	Trocas	442.266	Trocas	924.104
Quick Sort ITERATIVO	Passadas	6.709	Passadas	16.848	Passadas	33.596	Passadas	67.185
	Comparações	159.494	Comparações	402.578	Comparações	929.208	Comparações	2.103.882
	Trocas	88.396	Trocas	216.674	Trocas	520.436	Trocas	1.081.151
MELHOR RESULTADO	Passadas	Quick Sort Iterativo (6.709)	Passadas	Quick Sort Recursivo (16.815)	Passadas	Quick Sort Iterativo (33.596)	Passadas	Quick Sort Iterativo (67.185)
	Comparações	Quick Sort Iterativo (159.494)	Comparações	Quick Sor Iterativo (402.578)	Comparações	Quick Sort Iterativo (929.208)	Comparações	Quick Sort Iterativo (2.103.882)
	Trocas	Selection Sort (9.990)	Trocas	Selection Sort (24.984)	Trocas	Selection Sort (49.991)	Trocas	Selection Sort (99.982)

Considerações finais

Descreva, no quadro abaixo, suas considerações acerca do resultado da comparação dos algoritmos de ordenação.

Algoritmos de ordenação com uma complexidade mais baixa, como Bubble Sort e Selection Sort, utilizam menos recursos de memória e processamento durante sua execução. No entanto, esses algoritmos levam muito mais tempo para realizar a ordenação das listas, o que torna inviável sua utilização em listas maiores.

Já os algoritmos com um nível maior de complexidade, Merge Sort e Quick Sort, utilizam mais memória em sua execução, tornando a ordenação das listas mais rápida e mais eficiente.

Em relação às variáveis de estatística, determina-se que o algoritmo mais eficiente é aquele que realiza o menor número de Comparações, levando em consideração de que esse é o processo que mais utiliza recursos da máquina.