

Relatório de Algoritmos de Ordenação

Resultados completos baseados no arquivo: src/main/resources/static/results.csv

Resultados para o Dataset: aleatorio_com_rep

n	BubbleSort (ms)	InsertionSort (ms)	SelectionSort (ms)	MergeSort (ms)	HeapSort (ms)	QuickSort (ms)
1000	0.917	0.983	0.601	0.256	0.177	0.291
1600	2.788	0.329	1.138	0.231	0.262	0.194
2200	3.626	0.570	2.369	0.209	0.352	0.271
2800	6.206	0.809	3.151	0.276	0.481	0.370
3400	7.409	1.409	3.298	0.231	0.455	0.543
4000	8.693	1.818	5.502	0.414	0.713	0.560
4600	11.586	2.270	5.183	0.346	0.488	0.571
5200	15.787	3.086	6.398	0.529	0.583	0.598
5800	22.899	2.618	8.517	0.673	0.618	0.681
6400	23.508	3.061	10.013	0.751	0.904	0.889
7000	28.343	4.354	12.120	0.664	0.749	0.939

Resultados para o Dataset: aleatorio_sem_rep

n	BubbleSort (ms)	InsertionSort (ms)	SelectionSort (ms)	MergeSort (ms)	HeapSort (ms)	QuickSort (ms)
1000	2.107	0.928	0.566	0.263	0.169	0.248
1600	2.276	0.302	1.139	0.146	0.273	0.174
2200	4.388	0.652	1.848	0.211	0.359	0.259
2800	5.208	0.829	3.083	0.357	0.469	0.348
3400	10.270	0.536	3.446	0.374	0.439	0.506
4000	8.533	0.862	5.244	0.367	0.704	0.539
4600	10.836	1.915	7.268	0.400	0.499	0.448
5200	14.053	2.176	6.267	0.595	0.476	0.707
5800	19.337	2.952	6.542	0.540	0.421	0.729
6400	24.632	2.687	10.111	0.577	0.647	0.711
7000	29.776	3.044	11.373	0.645	0.731	0.864

Resultados para o Dataset: crescente_com_rep

n	BubbleSort (ms)	InsertionSort (ms)	SelectionSort (ms)	MergeSort (ms)	HeapSort (ms)	QuickSort (ms)
1000	0.466	1.090	3.427	1.454	1.771	2.036
1600	0.006	0.013	1.297	0.056	0.236	0.205
2200	0.004	0.007	2.331	0.014	0.315	0.157
2800	0.003	0.008	2.832	0.020	0.367	0.209
3400	0.005	0.012	5.209	0.043	0.377	0.288
4000	0.013	0.015	4.975	0.038	0.445	0.397
4600	0.005	0.013	5.450	0.057	0.487	0.388

5200	0.005	0.014	6.349	0.024	0.421	0.435
5800	0.009	0.018	7.372	0.037	0.495	0.411
6400	0.007	0.019	9.143	0.041	0.551	0.461
7000	0.007	0.022	12.106	0.065	0.659	0.566

Resultados para o Dataset: crescente_sem_rep

n	BubbleSort (ms)	InsertionSort (ms)	SelectionSort (ms)	MergeSort (ms)	HeapSort (ms)	QuickSort (ms)
1000	0.066	0.112	0.535	0.023	0.160	0.183
1600	0.003	0.005	1.240	0.025	0.192	0.143
2200	0.003	0.007	1.903	0.014	0.299	0.187
2800	0.006	0.012	1.936	0.030	0.503	0.295
3400	0.004	0.012	3.884	0.041	0.352	0.350
4000	0.005	0.022	7.139	0.052	0.559	0.437
4600	0.005	0.014	4.975	0.033	0.350	0.353
5200	0.011	0.019	8.842	0.055	0.529	0.550
5800	0.006	0.021	8.016	0.038	0.468	0.487
6400	0.008	0.026	13.136	0.041	0.598	0.684
7000	0.007	0.017	14.137	0.046	0.605	0.733

Resultados para o Dataset: decrescente_com_rep

n	BubbleSort (ms)	InsertionSort (ms)	SelectionSort (ms)	MergeSort (ms)	HeapSort (ms)	QuickSort (ms)
1000	4.809	3.249	7.749	0.584	0.215	0.371
1600	1.156	0.695	1.290	0.235	0.247	0.182
2200	1.943	1.174	2.390	0.096	0.486	0.184
2800	3.416	1.715	3.716	0.108	0.368	0.214
3400	6.919	2.089	3.846	0.155	0.361	0.306
4000	5.984	2.542	7.752	0.176	0.535	0.453
4600	14.371	3.471	9.787	0.315	0.450	0.419
5200	17.707	3.439	6.136	0.245	0.393	0.435
5800	20.830	4.066	7.205	0.246	0.451	0.414
6400	24.954	6.621	8.930	0.240	0.525	0.604
7000	31.178	4.492	13.127	0.300	0.527	0.499

Resultados para o Dataset: decrescente_sem_rep

n	BubbleSort (ms)	InsertionSort (ms)	SelectionSort (ms)	MergeSort (ms)	HeapSort (ms)	QuickSort (ms)
1000	3.632	1.790	0.557	0.895	0.153	0.170
1600	0.852	0.479	1.305	0.059	0.273	0.175
2200	2.822	0.988	2.379	0.098	0.357	0.237
2800	2.772	2.141	2.650	0.137	0.440	0.354
3400	7.494	2.478	2.480	0.133	0.346	0.374
4000	9.197	2.351	4.496	0.208	0.520	0.385

4600	11.211	2.854	6.708	0.201	0.399	0.593
5200	10.778	3.344	6.589	0.257	0.450	0.523
5800	21.033	3.748	12.553	0.278	0.560	0.612
6400	29.920	4.320	11.608	0.174	0.630	0.730
7000	27.898	4.944	10.318	0.359	0.589	3.482

Conclusão:

A análise dos resultados confirma a superioridade de desempenho dos algoritmos de complexidade $O(n \log n)$ (MergeSort, HeapSort, QuickSort) em comparação com os algoritmos de complexidade $O(n^2)$ (BubbleSort, InsertionSort, SelectionSort), especialmente para grandes volumes de dados. As diferenças de desempenho entre os datasets (crescente, decrescente, aleatório) também são notáveis, afetando principalmente o QuickSort (pior caso em dados ordenados) e o InsertionSort (melhor caso em dados quase ordenados).