

Comparativo entre tempo de execução de algoritmos de ordenação
 Adriano Zanella Junior
 Complexidade de Algoritmos – CAL

Tempo em microssegundos, utilizando a biblioteca “sys/time.h” do Windows, assim como nos exemplos apresentados pelo professor.

Execuções realizadas em processador Intel Core i5-2450M CPU @ 2.50GHz, sistema Windows 10 de 64bits, com 6GB de memória RAM.

Primeira execução em 06/09/2017:

Nº Elementos	Bubble Sort	Insert Sort	Merge Sort	Quick Sort com primeiro elemento como pivo	com elemento aleatório como pivo	Heap Sort	Counting Sort	Bucket Sort
Crescente								
25000	837640	823559	6313	874021	3133	5004	389	406
50000	3222419	3252011	10546	3486092	6307	10134	725	912
75000	7245892	7281296	17120	7846212	9857	15737	996	1015
100000	12808639	12963239	21259	14011649	12793	22850	1316	1934
1000000	1281806897	1311260196	227652	1410976203	211602	255096	20094	21242

Decrescente								
25000	778379	813530	5081	873117	3097	5045	385	411
50000	3159687	3233268	10785	3495570	6234	10460	648	703
75000	7082081	7255353	15802	7854702	9069	15763	1005	1008
100000	12500966	12943177	20261	13909531	12888	21948	1477	1321
1000000	1286047903	1319133793	226395	1421831393	200440	257888	19979	21753

Aleatório								
25000	936418	1148595	6609	3761	4139	5992	269	257
50000	3528289	4620112	13118	9470	10420	12240	506	491
75000	7735080	10383175	19914	16796	17240	19089	724	760
100000	13564904	18447502	27787	24756	27175	25777	1006	993
1000000	1329603899	1861537626	291105	1510502	1513615	318195	9452	9236

Aleatório com valor muito grande								
25000	937695	1150867	6575	3629	4083	5994	615062	598837
50000	3511379	4615090	12939	9202	9885	12455	630607	607320
75000	7737049	10356727	20251	16081	17166	19053	624284	609262
100000	13531616	18392595	26554	24865	26406	25853	636274	605931
1000000	1319471808	1866707352	289490	1509035	1506341	312946	632228	618819

O algoritmo Quick Sort é mais eficiente dos algoritmos com tempo não linear, embora que se o escolhido um pivô ruim, o tempo de execução aumenta. Enquanto os algoritmos lineares serem semelhantes, depende da implementação destes. Um valor muito fora do padrão nos algoritmos lineares também leva a uma perda de desempenho.

Segunda execução em 07/09/2017:

Nº Elementos	Bubble Sort	Insert Sort	Merge Sort	Quick Sort com primeiro elemento como pivo	Quick Sort com elemento aleatório como pivo	Heap Sort	Counting Sort	Bucket Sort
Crescente								
25000	788296	814083	6356	903589	2994	4951	356	477
50000	3178615	3259679	11401	3604229	6140	10101	665	714
75000	7299364	7309002	17589	8108439	9434	16010	1000	1027
100000	13186429	12991354	22213	14370927	12705	22027	1382	1771
1000000	1286306608	1318701552	234845	1502816100	193533	257572	20582	21065
Decrescente								
25000	786471	810342	5351	888059	3128	4968	346	401
50000	3131030	3253497	10203	3543931	6055	10179	665	722
75000	7096961	7309716	15812	7975540	10168	15865	990	1010
100000	12506293	13026298	21696	14194567	13078	21464	1292	1325
1000000	1274566351	1321090368	227860	1426852895	200299	254177	19822	20629
Aleatório								
25000	945190	1170625	7129	4469	4120	5915	277	270
50000	3522679	4615960	13421	9141	9948	12463	493	497
75000	7736040	10390344	20912	16252	17729	19140	727	752
100000	13549395	18478682	27544	25538	27201	26465	1005	999
1000000	1322475433	1984140811	293638	1510942	1531597	312439	9344	9116
Aleatório com valor muito grande								
25000	940861	1278934	6667	3671	4079	6063	615908	594820
50000	3495463	4892611	13326	9294	9848	12166	618540	602733
75000	7687032	11101521	20286	16297	17696	20034	611760	593999
100000	13504950	20448775	28473	24940	26176	26303	617421	599368
1000000	1323942299	1933217822	301287	1519572	1546676	313368	629174	602161