

Nr.Crt	Bubble	Count	Merge	Quick	Radix	Sort Implicit
<b>Testul 1</b>						
1	0.0008	0.0002	0.0004	0.0006	0.0003	0.0003
2	0.0007	0.0003	0.0005	0.0008	0.0003	0.0003
3	0.0005	0.0002	0.0005	0.0007	0.0003	0.0003
4	0.0005	0.0003	0.0006	0.0009	0.0003	0.0003
5	0.0007	0.0002	0.0006	0.0008	0.0005	0.0003
<b>Testul 2</b>						
1	0.0159	0.0008	0.0046	0.0049	0.0027	0.0028
2	0.0163	0.0006	0.0046	0.0046	0.0021	0.0035
3	0.016	0.0007	0.0046	0.0052	0.002	0.0028
4	0.0204	0.0006	0.0044	0.0045	0.002	0.0025
5	0.0198	0.0006	0.0047	0.0046	0.002	0.0027
<b>Testul 3</b>						
1	1.5005	0.0053	0.0545	0.051	0.0261	0.0351
2	1.5878	0.0058	0.0582	0.0501	0.0263	0.0341
3	1.5186	0.0055	0.0538	0.0497	0.0262	0.0337
4	1.5275	0.006	0.06	0.0499	0.0266	0.0332
5	1.6477	0.007	0.0532	0.0506	0.0309	0.0328
<b>Testul 4</b>						
1	151.005	0.0619	0.4215	3.9051	0.0898	0.1859
2	150.267	0.0644	0.4236	4.2432	0.0887	0.1788
3	150.297	0.065	0.4307	4.0032	0.1353	0.1904
4	148.829	0.0666	0.4315	3.9723	0.0837	0.1776
5	152.718	0.0627	0.4474	4.2688	0.0846	0.184
<b>Testul 5</b>						
1	164.128	0.1981	0.6401	0.5904	0.4522	0.4243
2	158.434	0.1988	0.6323	0.5872	0.4551	0.4083
3	156.679	0.1979	0.6309	0.5821	0.7318	0.4094
4	163.791	0.1982	0.6415	0.6115	0.4572	0.417
5	163.839	0.267	0.6311	0.609	0.4487	0.4224

<b>Medie -</b>						
<b>Testul 1</b>	0.00064	0.00024	0.00052	0.00076	0.00034	0.0003
<b>Medie -</b>						
<b>Testul 2</b>	0.01768	0.00066	0.00458	0.00476	0.00216	0.00286
<b>Medie -</b>						
<b>Testul 3</b>	1.55642	0.00592	0.05594	0.05026	0.02722	0.03378
<b>Medie -</b>						
<b>Testul 4</b>	150.6232	0.06412	0.43094	4.07852	0.09642	0.18334
<b>Medie -</b>						
<b>Testul 5</b>	161.3742	0.212	0.63518	0.59604	0.509	0.41628
<b>Medie</b>						
<b>Totala :</b>	62.714428	0.056588	0.225432	0.946068	0.127028	0.127312

*Testul 1 : 10 numere cu maximul 10*

*Testul 2 : 100 numere cu maximul 100*

*Testul 3 : 1000 numere cu maximul 1000*

*Testul 4 : 10000 numere cu maximul 10*

*Testul 5: 10000 numere cu maximul 10000*

*Concluzie : Sortarile ce au fost , in medie ( pe baza celor 5 teste facute ) , cele mai eficiente au fost Radix Sortul si Sortarea implicita a C++-ului. Urmatorul dupa ele ar fi Count Sortul. Se poate observa faptul ca, in cazul testului 4, unde au fost multe numere, insa foarte mici, countsort-ul a fost cel mai eficient. De altfel, in cazul unui numar mare de numere, BubbleSort-ul devine extrem de ineficient (Lucru observat prin cresterea exponentiala dintre testul 1 si testul 4 a timpului de executare ) , fiind recomandata utilizarea unei alte sortari. Toate informatiile acestea se bazeaza pe timpul de executare, ignorand complet eficienta din punctul de vedere al memoriei utilizate de fiecare algoritm.*