Tabele cu numărul de operații elementare

Vector sortat	100 elemente	300 elemente	10.000 elemente
Smooth Sort	855	29.301	984.617
Strand Sort	1.107	31.267	1.040.067
Shell Sort	230	18.926	945.302
Quicksort	585	452.985	500.099.985
Cocktail Sort	82	2.402	80.002
Bubble Sort	79	2.399	79.999
Insertion Sort	92	2.992	99.992

Vector generat aleator	100 elemente	300 elemente	10.000 elemente
Smooth Sort	1.208	104.842	5.569.859
Strand Sort	1.300	147.451	118.017.300
Shell Sort	468	39.744	2.763.202
Quicksort	256	22.590	1.326.338
Cocktail Sort	645	583.808	651.287.638
Bubble Sort	673	937.319	1.062.490.648
Insertion Sort	282	223.162	250.391.052

Vector sortat descrescător	100 elemente	300 elemente	10.000 elemente
Smooth Sort	1.484	104.196	4.841.796
Strand Sort	1.377	40.237	1.340.037
Shell Sort	412	30.490	2.247.274
Quicksort	485	362.985	400.099.985
Cocktail Sort	1.221	1.167601	1.299.920.001
Bubble Sort	1.267	12.124.52	1.349.915.002
Insertion Sort	542	451.492	500.049.992

Pentru Smooth Sort am ales implementarea care reprezintă implicit heapurile Leonardo în spațiu O(log n). Din cauza complexității calculelor, se poate observa că în cazul vectorului sortat crescător, numărul de operații elementare efectuate este considerabil mai mare decât în cazul celorlalți algoritmi care au cazul favorabil O(n) (cocktail, bubble, insertion).

Am ales ca și pivot pentru quicksort elementul de pe ultima poziție a partiției, așa cum este în cartea "Introduction to Algorithms". Din această cauză, complexitatea algoritmului este $O(n^2)$ atunci când rulează pe vectorii sortați (crescător sau descrescător), după cum se poate vedea și în numărul de operații efectuate.

Neașteptat de bine se comportă Shell Sort-ul, având cele mai bune performanțe per total. L-am implementat folosind gap-urile Ciura, prin urmare complexitatea defavorabilă nu se cunoaște.

În rest, după cum era de așteptat, Bubble Sort, Insertion Sort și Cocktail Sort merg foarte bine pe vectorii sortați crescător, și foarte slab pe vectorii sortați descrescător. Totuși, în cazul vectorului generat aleator de 10.000 de elemente se observă o îmbunătățire considerabilă de la Bubble Sort la Cocktail Sort.

Strand Sort-ul nu a fost implementat cel mai eficient cu putință, însă merge cel mai bine pe vectorul ordonat descrescător.

De reținut este faptul că deși Smooth Sort are o complexitate mai bună decât Shell Sort, datorită complexității implementării, efectuează un număr mai mare de operații elementare. Pentru a vedea diferența trebuie sortați vectori de mărime mult mai mare.