Algorithmics	Student information	Date	Number of session
	UO: 271407	23/02/2020	2
	Surname: Llera		
	Name: Elías		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Time measurements for sorting algorithms.

Bubble	times = 10		
n	sorted	inverse	random
10000	0 ms	79ms	127ms
20000	0 ms	281ms	516ms
40000	0 ms	1156ms	2109ms
80000	0 ms	4548ms	8485 ms
160000	0 ms	18267ms	34843 ms
320000	0 ms	72873ms	-
640000	0 ms	291881ms	-
1280000	16 ms	-	-
2560000	16 ms	-	-
5120000	16 ms	-	-
10240000	47 ms	-	-
20480000	94 ms	-	-
40960000	172 ms	-	-
81920000	328 ms	-	-
163840000	687 ms	-	-

These are the results obtained with the bubble algorithm, implemented with a sentinel. We can see that this improvement in the algorithm makes the times of the already sorted vectors go down in a very significant way, making it possible to take measures of very high workloads in almost no time. However, it is obvious that in the inverse and random scenarios, the executions times increase at a very high pace.

	Student information	Date	Number of session
Algorithmics	UO: 271407	23/02/2020	2
	Surname: Llera		
	Name: Elías		

Insertion	times = 10		
n	sorted	inverse	random
10000	16 ms	79ms	47 ms
20000	0 ms	359 ms	188 ms
40000	0 ms	547 ms	265 ms
80000	0 ms	2063 ms	1031 ms
160000	0 ms	8219 ms	4093 ms
320000	0 ms	32843 ms	16359 ms
640000	0 ms	131978 ms	65568 ms
1280000	0 ms	-	-
2560000	0 ms	-	-
5120000	16 ms	-	-
10240000	47 ms	-	-
20480000	94 ms	-	-
40960000	188 ms	-	-
81920000	359 ms	-	-
163840000	734 ms	-	-

These were the results of the Insertion algorithm. We can see that the results of the already sorted vector are very good, as already expected. This is the best algorithm for the mentioned scenario. However, this is an unlucky case to have, and we see that the times for the inverse and random, although they are better than the ones of the bubble algorithm, are still considerably big and increase at a high pace.

	Student information	Date	Number of session
Algorithmics	UO: 271407	23/02/2020	2
	Surname: Llera		
	Name: Elías		

selection	times = 10		
n	sorted	inverse	random
10000	187ms	579 ms	377 ms
20000	656 ms	1687 ms	1443 ms
40000	2579 ms	6736 ms	5720 ms
80000	10188 ms	26929 ms	23082 ms
160000	40749 ms	107597 ms	91400 ms
320000	163259 ms	430341 ms	-
640000	652221 ms	1720438ms	-

In the selection algorithm, we can see that the times are quite bad in every category. They are high and increase at a very high speed, making it, again, a not very usable algorithm.

Algorithmics	Student information	Date	Number of session
	UO: 271407	23/02/2020	2
	Surname: Llera		
	Name: Elías		

quicksort	times = 1		
n	sorted	inverse	random
10000	42ms	31 ms	4 ms
20000	63 ms	78 ms	4 ms
40000	-	-	8 ms
80000	-	-	7 ms
160000	-	-	14 ms
320000	-	-	29 ms
640000	-	-	62 ms
1280000	-	-	132 ms
2560000	-	-	273 ms
5120000	-	-	573 ms
10240000	-	-	1219 ms
20480000	-	-	2582 ms
40960000	-	-	5273 ms
81920000	-	-	11580 ms
163840000	-	-	26369 ms

In the quicksort algorithm, we can see the times being much better than the others. Although the selection of the pivot is not the best, we can see that the workload doesn`t have in the execution time the impact than the formers ones had. We can see the random category (the most common scenario) being able to work with very high workloads in an acceptable time. However, these measures cannot be compared with the others directly. Since the algorithm requires more memory because of the recursive calls, the nTimes parameter was reduced to 1, in an attempt to get as many values as possible. We can see this working for the random category, but the sorted and inverse one could only produce 2 values before launching a Stack Overflow Exception.

CPU: Intel® Core™ i5-3470 CPU @ 3.20GHz

RAM: 8GB