Algorithmics	Student information	Date	Number of session
	UO:UO277653	25/02/21	4
	Surname: Stanci		
	Name: Stelian Adrian		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Time measurements for sorting algorithms

All the times are expressed in ms, as the number of times the algorithms are repeated is just 1.

Insertion

N	Sorted(t)	Inverse(t)	Random(t)
10000	1	107	52
20000	1	313	188
40000	2	371	153
80000	1	1289	779
160000	2	5427	2372
320000	0	21113	9477
…			
81920000	35	-	-
163840000	69	-	-
327680000	137	-	-

We can observe that in insertion, when the elements are already sorted, the execution is very fast, as it is not necessary to make any changes and it just iterates through the array. We can see that in the last 3 iterations there is a linear trend, which is the best complexity we can obtain with this algorithm.

In inverse and random we can see the most likely trend of this algorithm, which is quadratic. For example, in the iteration from 160000 to 320000 the size of the problem is multiplied by 2, but the time obtained is multiplied by 4.

Escuela de Ingeniería Informática

Algorithmics	Student information	Date	Number of session
	UO:UO277653	25/02/21	4
	Surname: Stanci		
	Name: Stelian Adrian		

Selection

N	Sorted(t)	Inverse(t)	Random(t)
10000	27	57	35
20000	79	206	100
40000	278	824	354
80000	987	2711	1275
160000	3479	6674	3819
320000	13175	24315	14217

There is a very noticeable quadratic trend in the three scenarios. We can also note that it is faster in the sorted version than in the other ones, and that the worst is the inverse version. Again, in the step from an n to 2*n the time is multiplied by 4, as seen for example in the last two rows.

Bubble

N	Sorted(t)	Inverse(t)	Random(t)
10000	67	95	234
20000	261	317	937
40000	349	1227	2831
80000	3856	4710	11054
160000	13894	18620	45432
320000	54179	77833	172638

The worst algorithm so far in terms of speed. It performs the worst with a random vector, and the best with a sorted one. It also has a quadratic trend. With the time growing 4 times when the size doubles, as with the previous ones.

Algorithmics	Student information	Date	Number of session
	UO:UO277653	25/02/21	4
	Surname: Stanci		
	Name: Stelian Adrian		

Quicksort (central element as pivot)

N	Sorted(t)	Inverse(t)	Random(t)
10000	3	2	3
20000	1	2	5
40000	4	4	11
…			
10240000	357	480	1681
20480000	559	793	3582
40960000	1165	1710	7580

Here we can notice a very big improvement in terms of speed as a result of the complexity this algorithm has O(nlogn), which is way better than the one of the previous ones (O(n^2), except insertion, whose complexity in the best case can be O(n)).

Activity 2. QuicksortFateful

The pivot selected is the first element of the partition, which can lead to some performance problems, as we risk forming a list as we do the partitions, and thus that would overall lead to a complexity of O(n^2), far worse than the average one.

The only scenario where this would work is if, by any chance, we select the value which is the closest to the median, any other case would lead to a worse performance as a result of not dividing the workload correctly.

Using the median of 3 we get a value which is very good for dividing the workload in half and thus obtaining a O(nlogn) complexity, but with this strategy of selecting the first number is very unlikely to get near that.