Algorithmics	Student information	Date	Number of session
	UO: 293079	1/3/22	2
	Surname: GAVALAS		
	Name: DIMITIROS		

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Time measurements for sorting algorithms.

Insertion

n	Sorted()	inverse(t)	random(t)
10000	0	76	97
20000	0	164	168
40000	3	269	247
80000	1	1040	983
160000	2	3950	3976
320000	1	14887	
640000	0		
128000	1		

1. Best=values of array are in order. Complexity = O(n)
2. Worst=values of array are not in order. Complexity = O(n^2)
3. Values of array are random. Complexity = O(n^2)

Algorithmics	Student information	Date	Number of session
	UO: 293079	1/3/22	2
	Surname: GAVALAS		
	Name: DIMITIROS		

Selection

n	Sorted()	inverse(t)	random(t)
10000	29	68	43
20000	82	145	135
40000	326	576	583
80000	1313	2613	2050
160000	5160		8043
320000	20947		
640000	82153		
128000			

1. Best=sorted. Complexity = O(n^2)
2. Worst =not sorted. Complexity = O(n^2)
3. Values of array are random. Complexity = O(n^2)

All cases have quadratic complexity, since the algorithm needs to find the position of the smallest O(n) and swap: O(n) * O(n) == O(n2)

Algorithmics	Student information	Date	Number of session
	UO: 293079	1/3/22	2
	Surname: GAVALAS		
	Name: DIMITIROS		

Bubble

n	Sorted()	inverse(t)	random(t)
10000	41	168	161
20000	114	721	720
40000	423	3060	2996
80000	1659	12091	11999
160000	6458		
320000			
640000			
128000			

1. Best=sorted. Complexity = O(n^2)
2. Worst=random. Complexity = O(n^2)
3. Values of array are in reverse. Complexity = O(n^2)

All cases have quadratic complexity, since you have to iterate over the entire array

Algorithmics	Student information	Date	Number of session
	UO: 293079	1/3/22	2
	Surname: GAVALAS		
	Name: DIMITIROS		

Quicksort

n	Sorted()	inverse(t)	random(t)
10000	1	1	1
20000	2		2
40000	4		6
80000	1		
160000			
320000			
640000			
128000			

StackOverflow starting at n = 10000

1. Best=sorted. Complexity = O(nlogn)

2. Worst=random. Complexity = O(n^2)

3. Values of array are in reverse. Complexity = O(nlogn)

Best algorithm between our 4 options.

Algorithmics	Student information	Date	Number of session
	UO: 293079	1/3/22	2
	Surname: GAVALAS		
	Name: DIMITIROS		

CONCLUSION. Quicksort is much faster than rest of the algorithms that we have used.

Activity 2. QuicksortFateful.

Briefly explain what the criteria is for selecting the pivot in that class. Indicate when that idea can work and when that idea will not work.

When choosing the first or last array element as the pivot, then Quicksort takes n^2 time to sort an already-sorted array.
Always choosing first or last would cause worst-case performance for nearly-sorted or nearly-reverse-sorted data.
If the items of the array are in correct order in relation to the pivot we dont have to compare elements because we know they are in the correct spot in relation to the pivot.