Algorithmics	Student information	Date	Number of session
	UO: 275725		2
	Surname: Gómez Menéndez		
	Name: Laura		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. [Time measurements for sorting algorithms.]

Four tables with times for each of the algorithms (Insertion, Selection, Bubble and Quicksort with the central element as the pivot).

A brief explanation (a paragraph) for each of the tables to conclude whether the values make sense regarding the expected theoretical complexity.

Insertion

N	sorted(t) (nTimes=1000)	inverse (t)(nTimes=1)	random(t) (nTimes=1)
10000		141	81
20000		559	287
40000	53	699	353
80000	127	3061	1426
160000	212	14017	61260
320000	501	48078	23857
640000	1225		291074
1280000	2473		

The complexity of Insertion is O(n^2) at its worst case and O(n) at its best case. If we try to calculate the execution time of 80000 (sorted):

If the complexity is O(n^2):

K = 80000/20000 = 4

T2 = k^c * t1 = 4^2 * 559 = 8944 and my measurement is 3061, so there is a big difference.

Algorithmics	Student information	Date	Number of session
	UO: 275725		2
	Surname: Gómez Menéndez		
	Name: Laura		

If the complexity is O(n2):

K = 80000/20000 = 4

T2 = k * t1 = 4 * 559 = 2236 and my measurement is 3061, so it is more or less ok.

Selection (nTimes=1)

N	sorted(t)	inverse (t)	random(t)
10000	35	83	201
20000	131	254	806
40000	524	1024	3680
80000	1356	4273	14768
160000	5475	19305	57254
320000	22741	64338	
640000	95466		

The complexity of Selection is O(n^2) at its best and worst case. If we try to calculate the execution time of 80000 (sorted):

K = 80000/20000 = 4

T2 = k^c * t1 = 4^2 * 131 = 2096 and my measurement is 1356 so there is a little difference but not too big.

Algorithmics	Student information	Date	Number of session
	UO: 275725		2
	Surname: Gómez Menéndez		
	Name: Laura		

Bubble(nTimes=1)

N	sorted(t)	inverse (t)	random(t)
10000	175	192	172
20000	625	679	625
40000	960	2570	960
80000	10185	11074	10185
160000	34875	38912	34872
320000	86304	157373	86304

The complexity of Bubble is O(n^2) at its best and worst case. If we try to calculate the execution time of 80000 (sorted):

K = 80000/20000 = 4

T2 = k^c * t1 = 4^2 * 625 = 10000 and my measurement is 10185 so it´s ok.

Algorithmics	Student information	Date	Number of session
	UO: 275725		2
	Surname: Gómez Menéndez		
	Name: Laura		

Quicksort(nTimes=1)

N	sorted(t)	inverse (t)	random(t)
640000		94	110
1280000		203	203
2560000	63	406	391
5120000	140	875	797
10240000	266	1594	1593
20480000	594	3875	3359
40960000	1062	6688	6829
81920000	2188		14093

The complexity of Quicksort is O(nlogn) at its best and O(n^2) at its worst case, so it have an average of O(nlogn). If we try to calculate the execution time of 20480000(inverse):

-Assuming O(nlogn)

K = 20480000/5120000 = 4

T2 = k*(logn2/logn1)*t1=4*(log20480000/log5120000)*875=3814.07 and the time that I measured is 3875 which is very similar so we can say that its complexity is O(nlogn).

We can also realize that if we compare this one with the others, we can see that it is so much faster.

Algorithmics	Student information	Date	Number of session
	UO: 275725		2
	Surname: Gómez Menéndez		
	Name: Laura		

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.

The pivot that is selected under this criteria is the first one, the one that is at the left and then it interchange it with its right position.

1._ If the number than we are comparing is smaller or equal than the pivot and its position is less than its right we assume that the first element is bigger than the pivot.

2._ If the number than we are comparing is bigger than the pivot and its position is bigger than its left we assume that the first element is smaller than the pivot.

If the number of the first option is smaller than the second one we interchange both.

It is no really good because if we choose the right or left one, we can find the smaller or de bigger, so the complexity increases.