Algorithmics	Student information	Date	Number of session
	UO: 294665	27/02/2024	3
	Surname: García Castro		
	Name: Gonzalo		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. [Bubble Algorithm]

n	T ordered	T reverse	T random
10000	0,621	1,998	1,361
2*10000	2,297	8,050	8,767
2**2*10000	9,340	30,635	30,246
2**3*10000	37,998	Oot	Oot
2**4*10000	Oot	Oot	Oot

The complexity of the bubble algorithm is O(n^2) as studied in theory lessons. Bubble algorithm works well with small sets but it is not so efficient. We can see how the complexity is increasing exponentially with the iterations.

Activity 2. [Selection Algorithm]

n	T ordered	T reverse	T random
10000	0,945	0,537	0,544
2*10000	2,206	2,116	1,931
2**2*10000	11,715	8,746	7,587
2**3*10000	44,892	34,768	30,959
2**4*10000	Oot	Oot	Oot

The complexity of the selection algorithm is O(n^2) as studied in theory lessons. We can see that it goes exponentially higher and higher until Oot. It is useful for not ordering all the list, but for ordering parts and getting the bigger element in a certain list.

Algorithmics	Student information	Date	Number of session
	UO: 294665	27/02/2024	3
	Surname: García Castro		
	Name: Gonzalo		

Activity 3. [Insertion Algorithm]

n	T ordered	T reverse	T random
10000	LoR	0,97	0,375
2*10000	LoR	3,793	1,449
2**2*10000	LoR	11,532	5,848
2**3*10000	LoR	47,513	24,324
2**4*10000	LoR	Oot	Oot
…			
2**13*10000	1,875	Oot	Oot

The complexity for this algorithm is O(n^2) in the worst case and O(n) in the best case. We can see how fast it is in the best case when it is ordered, but when it's reverse it is worse as it has to go through all the elements in the array. But when they are randomly distributed, it is kind of mixed but is anyway O(n^2).

Algorithmics	Student information	Date	Number of session
	UO: 294665	27/02/2024	3
	Surname: García Castro		
	Name: Gonzalo		

Activity 4. [Quicksort Algorithm]

n	T ordered	T reverse	T random
250000	0,05	0,059	0,280
2*250000	0,101	0,114	0,718
2**2*250000	0,212	0,232	1,714
2**3*250000	0,434	0,484	3,633
2**4*250000	0,91	0,999	5,230
2**5*250000	1,874	2,101	12,224
2**6*250000	3,867	4,262	15,674

As studied in theory lessons, the quicksort algorithm has a complexity of O(n*log(n)) in the best and O(n^2) in the worst case. We can see the complexity is reflected in the previous table, with O(n*log(n)) in the ordered and reverse cases. We can see that the complexity is the same but when it is randomly ordered, but it takes longer as it is a real case. We can see how powerful is quicksort algorithm compared with the other ones.

Algorithmics	Student information	Date	Number of session
	UO: 294665	27/02/2024	3
	Surname: García Castro		
	Name: Gonzalo		

Activity 5. [QuicksortInsertion Algorithm]

N	T random
Quicksort	17,330
Quicksort+Insertion (k=5)	17,213
Quicksort+Insertion (k=10)	17,019
Quicksort+Insertion (k=20)	16,726
Quicksort+Insertion (k=30)	16,516
Quicksort+Insertion (k=50)	16,100
Quicksort+Insertion (k=100)	14,619
Quicksort+Insertion (k=200)	12,246
Quicksort+Insertion (k=500)	18,765
Quicksort+Insertion (k=1000)	37,318

Quicksort with insertion is a really good algorithm as it combines the best part of quicksort (the best option for big algorithms) and insertion (works really well with small and ordered algorithms). So, as we can see in the above table, it lasts less until we reach k=200 and then it grows exponentially. That's because when we reach a certain point (in this case 200) the problem gets so bigger that the algorithm is not good enough to be fast.

The times are taking from a laptop with

Processor: Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz   1.19 GHz

RAM: 16,0 GB