Algorithmics	Student information	Date	Number of session
	UO: 271407	18/02/2020	1.2
	Surname: Llera		
	Name: Elías		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Two algorithms with the same complexity

N	loop2(t)	loop3(t)	loop2(t)/loop3(t)
1	0 ms	0 ms	-
2	0 ms	0 ms	-
4	0 ms	0 ms	-
8	0 ms	0 ms	-
16	0 ms	0 ms	-
32	16 ms	0 ms	-
64	16 ms	16 ms	1
128	93 ms	47 ms	1,978723404
256	360 ms	187 ms	1,92513369
512	1453 ms	734 ms	1,979564033
1024	5792 ms	2869 ms	2,018821889
2048	22915 ms	11466 ms	1,998517356
4096	91745 ms	45845 ms	2,001199695

According to the measures, we can see that, although loop2 takes about twice as long than loop3 to finish, the result of the division is always about the same. This is because, despite the operations taken have more weight in loop2, the complexity of both algorithms is the same, so the time taken increases at the same rhythm.

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

RAM: 8GB

Algorithmics	Student information	Date	Number of session
	UO: 271407	18/02/2020	1.2
	Surname: Llera		
	Name: Elías		

Activity 2. Two algorithms with different complexity

N	loop1(t)	loop2(t)	loop1(t)/loop2(t)
1	0 ms	0 ms	-
2	0 ms	0 ms	-
4	0 ms	0 ms	-
8	0 ms	0 ms	-
16	0 ms	0 ms	-
32	0 ms	16 ms	0
64	0 ms	16 ms	0
128	0 ms	93 ms	0
256	16 ms	360 ms	0,044444444
512	31 ms	1453 ms	0,021335169
1024	63 ms	5792 ms	0,010877072
2048	140 ms	22915 ms	0,006109535
4096	297 ms	91745 ms	0,003237234
8192	625 ms	-	-
16384	1357 ms	-	-

These results make sense with the complexities, since loop1 (nlogn complexity) has a much shorter execution time than loop2(n^2 complexity). The big difference in the times taken made these algorithms hard to compare, since some measures where to short to be considered in loop1 and others too long in loop2. However, the difference in performance is very clear, loop1 is much faster.

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

RAM: 8GB

Algorithmics	Student information	Date	Number of session
	UO: 271407	18/02/2020	1.2
	Surname: Llera		
	Name: Elías		

Activity 3. Complexity of other algorithms

N	loop4(t)	loop5(t)	loop4(t)/loop5(t)
1	0 ms	0 ms	-
2	0 ms	0 ms	-
4	0 ms	0 ms	-
8	0 ms	0 ms	-
16	0 ms	0 ms	-
32	0 ms	0 ms	-
64	16 ms	16 ms	1
128	218 ms	116 ms	1,879310345
256	3094 ms	915 ms	3,381420765
512	44778 ms	7698 ms	5,816835542
1024	598434 ms	66031 ms	9,06292499

For these algorithms, taking measures with good times was a bit more complicated. For low n, they take very short times, but as soon as the workloads increase the times increase very fast as well. We can see that as soon as we start having time measures (n = 128), the time of loop5 is a little bit les than half of loop4. This difference increases the more n increases, since n^4 increases much faster than $n^3 \log n$.

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

RAM: 8GB

Algorithmics	Student information	Date	Number of session
	UO: 271407	18/02/2020	1.2
	Surname: Llera		
	Name: Elías		

Activity 4. Study of Unknown.java

N	Unknown(t)
1	0 ms
2	0 ms
4	0 ms
8	0 ms
16	0 ms
32	0 ms
64	16 ms
128	15 ms
256	110 ms
512	828 ms
1024	5331 ms
2048	36808 ms
4096	265946 ms

This algorithm as cubic complexity, since it has three nested loops all of them depending on the workload. At first sight, this seems a possibility since the times (in ms) taken increase very quickly with N. Lets calculate the theoretical results taking as a reference N = 512 and t = 828. According to the theoretical expression, we have that $t_2=(n_2^2/n_1^2) * t_1$. For n = 1024 that is t = 6624 and for n = 2048, t = 52992. These measures are clearly far away from the ones taken.

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

RAM: 8GB