Algorithmics	Student information	Date	Number of session
	UO:276903	23/2/21	2
	Surname: Garriga Suárez		
	Name: Carlos		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Two algorithmics with the same complexity

N	loop2(t)	loop3(t)	loop2(t)/loop3(t)
8	0	0	0
16	0	0	0
32	1	0	0
64	1	1	1
128	2	1	2
256	8	4	2
512	24	13	1,84
1024	97	49	1.97
2048	382	191	2
4096	1531	765	2
8192	6213	3062	2,02
16384	24513	12207	2,01

Tens of milliseconds, nTimes = 10. Both algorithms had the same complexity which was O(n^2). Anyway, we get very different results because their behavior is not the same. We can observe that loop 2 has two nested loops which go until the parameter n but, in loop 3 the second loop only reaches the variable of the first loop. This is that the loop 3 do half iterations that loop 2 does resulting in half time. That is why we get an implementation constant 2 between loop 2 and loop 3.

Activity 2. Two algorithmics with different complexity

N	loop1(t)	loop2(t)	loop1(t)/loop2(t)
8	0	0	0
16	0	0	0
32	1	1	1
64	1	1	1
128	1	2	0,5
256	1	8	0,125
512	1	24	0,0416
1024	1	97	0,01
2048	3	382	0,007
4096	5	1531	0,003
8192	11	6213	0,001
16384	23	24513	0,0009

Algorithmics	Student information	Date	Number of session
	UO:276903	23/2/21	2
	Surname: Garriga Suárez		
	Name: Carlos		

Tens of milliseconds, nTimes = 10. In this case the complexity of the algorithms is not the same so we will not have an implementation constant. The loop 1 has a O(nlogn) complexity and the loop 2 has a O(n^2). Makes sense from the point of view that the first algorithm spends less time than the second. We have a tendency in the division to 0, that means that the first algorithm is better than the second one.

Activity 3. Complexity of other algorithms

N	*loop4(t)*	*loop5(t)*	*loop4(t)/loop5(t)*
1	0	0	0
2	0	0	0
4	0	0	0
8	1	1	1
16	1	1	1
32	12	4	1
64	153	18	3
128	2429	152	15,98
256	38777	1363	28,44

Milliseconds, nTimes = 1. Tens of milliseconds, nTimes = 10. In this case the complexity of the algorithms is not the same so we will not have an implementation constant. The loop 4 has a O(n^4) complexity and the loop 5 has a O(n^3logn). From the theoretical point of view the second algorithm is better than the first one. We can see in the measurement that it is true as the second algorithm lasts less than the first one. We can also see that the ratio of the division tends to infinity whose meaning is that the second algorithm is better.

Activity 4. Study of Unknown.java

N	*Unknown(t)*	*Theoretical measurement*
8	0	0
16	1	1
32	3	8
64	19	64
128	145	512
256	942	4096
512	6590	32768
1024	49322	262144

Microseconds. The complexity of this algorithm is O(n^3). We can observe that the theoretical results do not make sense with the measurements. That is because the unknown algorithm are three nested loops whose two most internals do not reach n amount of work.

```
for (int i=1; i<=n; i++)
    for (int j=1; j<=i; j++)
        for (int k=1; k<= j; k++)
```

The second loop only reaches until i and the third one only reaches j. Therefore, it is doing less iterations than if the three loops reached n.