Algorithmics	Student information	Date	Number of session
	UO: 293860	13/02/2024	3
	Surname: López Álvarez		
	Name: Juan		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Table 1 iterative models

TABLE 1				
N	tLoop1	tLoop2	tLoop3	tLoop4
100	49	36	91	2
200	104	131	334	14
400	226	610	1253	116
800	521	2793	5407	643
1600	1124	11096	23590	5009
3200	2432	52214	Oot	39197
6400	5126	Oot	Oot	Oot
12800	13184	Oot	Oot	Oot
25600	25479	Oot	Oot	Oot
51200	53325	Oot	Oot	Oot

The complexity of the first program Loop1 is O(n) as it is composed by a single loop that goes from 1 to 2*n with step 3. If we simplify the constants we get that the complexity is n. The results obtained demonstrate our computations as we can see that the relation between the size of the problem and the time is linear. The second's program complexity is $O(n^2 \log_3 n)$ as it is composed by two nested loops with complexity n inside a while loop with complexity $\log_3 n$. If we make the comparison in excel we can see that the graphs obtained are very similar. The third program has a complexity of n^2 $\log_2$ n, it is very similar to the previous one but in this case the while loop has complexity n and contains two nested loops, one with complexity n and another one with complexity $\log_2 n$. The last one has complexity n^3 because it has three nested loops with complexity n. The results obtained match the complexities explained before (more or less).

Algorithmics	Student information	Date	Number of session
	UO: 293860	13/02/2024	3
	Surname: López Álvarez		
	Name: Juan		

Activity 2. Creation of iterative models of a given time complexity

TABLE 2			
N	tLoop5	tLoop6	tLoop7
100	10	3	0
200	50	16	1
400	181	157	17
800	851	1382	256
1600	4382	11782	3615
3200	19851	Oot	55496
6400	Oot	Oot	Oot

The times obtained match the complexities supposed, the loop5 was implemented with 4 loops, two of them increase n by one and two of them multiplying I by 2. The loop6 and loop7 algorithms were implemented in a similar way but changing the number of loops that increase I with a multiplication.

Activity 3. Two algorithms with different complexity

TABLE 3			
N	tLoop1	tLoop2	t1/t2
100	10	336	0,0297619
200	24	1240	0,01935484
400	47	6002	0,00783072
800	114	26931	0,00423304
1600	245	Oot	
3200	527	Oot	
6400	1089	Oot	
12800	2446	Oot	
25600	5123	Oot	
51200	10776	Oot	

The first loop is way better than the second one because the relation is lower than 1 and quite close to 0, this was expected because with the same number of repetitions the Loop2 gave times > 60s for a size >1600.

Algorithmics	Student information	Date	Number of session
	UO: 293860	13/02/2024	3
	Surname: López Álvarez		
	Name: Juan		

Activity 4. Two algorithms with the same complexity

TABLE 4			
N	tLoop2	tLoop3	t1/t2
100	336	1171	0,28693424
200	1240	5193	0,23878298
400	6002	22644	0,26505918
800	26931	Oot	
1600	Oot	Oot	
3200	Oot	Oot	
6400	Oot	Oot	
12800	Oot	Oot	
25600	Oot	Oot	
51200	Oot	Oot	

According to the theoretical time complexity, both algorithms should present similar execution times, however we see that the Loop3 is much slower than Loop2, this could be because the constants used in the loops are different, which don't affect the complexity, but they do affect the execution time.

Activity 5. Same algorithm in different development environments

TABLE 5						
N	tLoop4 (Python) - t41	tLoop4 (Java w/o optimization) - t42	tLoop4 (Java w/o optimization) - t42 x10	tLoop4 (Java w/ optimization) - t43 x10	t42/t41	t43/t42
100	3	1	13	3	0,33333333	0,23076923
200	26	8	86	2	0,30769231	0,02325581
400	218	63	655	7	0,28899083	0,01068702
800	1807	469	4693	30	0,25954621	0,0063925
1600	15186	3788	37489	174	0,24944027	0,00464136
3200	Oot	29858	Oot	1269	-	-
6400	Oot	Oot	Oot	8310	-	-

In this occasion we see that the even when the algorithm is the same the environment is very important, in python there is no compiler so the evaluation of the code is done runtime, this is the reason of the lower performance in comparison with the java version, which is compiled and later run.