Algorithmics	Student information	Date	Number of session
	UO: UO294786	14/02/24	2
	Surname: Álvarez Iglesias		
	Name: Rafael		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. [SOME ITERATIVE MODELS]

YOU ARE REQUESTED TO: After measuring times, fill in the table:

n	Time(ms)			
	Loop1	Loop2	Loop3	Loop4
100	0,0106	0,354	1,34	1,16
200	0,0214	1,266	5,94	8,56
400	0,0486	5,956	25,03	64,77
800	0,1126	27,452	106,22	499,11
1600	0,249	107,872	454,79	3940,74
3200	0,5278	488,166	1973,37	OoT
6400	1,2056	OoT	OoT	OoT
12800	2,4776	OoT	OoT	OoT
25600	5,3678	OoT	OoT	OoT
51200	10,996	OoT	OoT	OoT

Explain whether the different times obtained agree with what was expected, according to the theoretical complexity of the four cases:

By calculating the complexity of each algorithm, the following results were obtained:

- Loop1 has a complexity O(n * log(n)).
- Both Loop2 and 3 has a complexity O(n^2 * log(n)).
- Loop4 has a complexity O(n^3).

After analyzing the obtained results, all of them follow their expected trend. Both algorithms with O(n^2*log(n)) complexity grow in a similar manner. Loop4 is the slowest of them all, as expected due to it having the worst complexity. This increase in the complexity of the algorithms can be indicated by the amount of Out Of Time executions in each one of the algorithms. Loop1 was able to execute for each of the n cases, whereas both 2 and 3 had to be paused at the same n, and Loop4 had to end at a lower value.

Algorithmics	Student information	Date	Number of session
	UO: UO294786	14/02/24	2
	Surname: Álvarez Iglesias		
	Name: Rafael		

Activity 2. [CREATION OF ITERATIVE MODELS OF A GIVEN TIME COMPLEXITY]

<u>YOU ARE REQUESTED TO:</u> Implement three new classes Loop5.java, Loop6.java and Loop7.java, that simulate iterative algorithms with a complexity O(n2log2n), O(n3log n) and O(n4) respectively.

After implementing the classes, measure their execution times and fill in the table:

n	Time(ms)		
	Loop5	Loop6	Loop7
100	3,26	42,4	330
200	16,3	368,6	5171
400	78,92	3239	81283
800	378,34	28031,6	OoT
1600	1776,16	OoT	OoT
3200	OoT	OoT	OoT
6400	OoT	OoT	OoT
12800	OoT	OoT	OoT
25600	OoT	OoT	OoT
51200	OoT	OoT	OoT

Explain whether the different times obtained agree with what was expected, according to the theoretical complexity of the four cases:

All of the studied algorithms behave according to their complexity. As it was expected, Loop7 with a complexity O(n^4) took the most time to execute, with a very big growth in time each execution. Between Loop5 and Loop6, Loop6 shows a greater increase in time due to it's higher exponent multiplying the logarithmic trend. It is safe to assume that both of them grow as expected.

Algorithmics	Student information	Date	Number of session
	UO: UO294786	14/02/24	2
	Surname: Álvarez Iglesias		
	Name: Rafael		

Activity 4. [COMPARISON OF TWO ALGORITHMS: TWO ALGORITHMS WITH DIFFERENT COMPLEXITY]

YOU ARE REQUESTED TO: After measuring times, fill in the table:

n	Time(ms)		
	Loop1	Loop2	t1/t2
100	0,0106	0,354	0,0299435
200	0,0214	1,266	0,01690363
400	0,0486	5,956	0,00815984
800	0,1126	27,452	0,0041017
1600	0,249	107,872	0,00230829
3200	0,5278	488,166	0,00108119
6400	1,2056	OoT	-
12800	2,4776	OoT	-
25600	5,3678	OoT	-
51200	10,996	OoT	-

Explain whether the different times and their quotient agree with what was expected according to the theoretical complexity:

The obtained quotient is always lower than 1. That indicates that Loop1 complexity is better than Loop2's. That is, in fact, what it's expected from their complexities given that Loop1 has O(n * log (n)), whereas Loop2 has O(n^2 * log(n)), which is undoubtedly worse.

Algorithmics	Student information	Date	Number of session
	UO: UO294786	14/02/24	2
	Surname: Álvarez Iglesias		
	Name: Rafael		

Activity 5. [COMPARISON OF TWO ALGORITHMS: TWO ALGORITHMS WITH THE SAME COMPLEXITY]

<u>YOU ARE REQUESTED TO:</u> After measuring times, fill in the table:

n	Time(ms)		
	Loop3	Loop2	t3/t2
100	1,34	0,354	3,78531073
200	5,94	1,266	4,69194313
400	25,03	5,956	4,20248489
800	106,22	27,452	3,86929914
1600	454,79	107,872	4,21601528
3200	1973,37	488,166	4,0424159
6400	OoT	OoT	-
12800	OoT	OoT	-
25600	OoT	OoT	-
51200	OoT	OoT	-

Explain whether the different times and their quotient agree with what was expected according to the theoretical complexity:

The obtained quotient is always higher than 1. That indicates that Loop3 complexity is worse than Loop2's. That is not to be expected. With both algorithms supposedly having the same complexity O(n^2 * log(n)), the quotient should be closer to 1.

Algorithmics	Student information	Date	Number of session
	UO: UO294786	14/02/24	2
	Surname: Álvarez Iglesias		
	Name: Rafael		

Activity 6. [COMPARISON OF TWO ALGORITHMS: SAME ALGORITHM IN DIFFERENT DEVELOPMENT ENVIRONMENTS]

<u>YOU ARE REQUESTED TO:</u> After measuring times, fill in the table:

n	Time(ms)				
	Python (t41)	Java NO OPT (t42)	Java OPT (t43)	t42/t41	t43/t42
200	59	8,56	0,626	0,14508475	0,07313084
400	351	64,77	2,828	0,18452991	0,04366219
800	2892	499,11	13,436	0,17258299	0,02691992
1600	23470	3940,74	62,756	0,16790541	0,01592493
3200	211462	OoT	281,952	-	-
6400	OoT	OoT	OoT	-	-

Explain whether the different times and their quotient agree with what was expected according to the theoretical complexity:

As expected, comparing Java, even without optimizations, against Python, leads to a quotient lower than 1. That is, Java is faster than Python. When comparing Java with and without JIT, a quotient lower than 1 is obtained once again. This case, Java with optimizations is faster than Java without them, as it was expected. All these results are in accordance with what was expected, even though the complexity itself for the algorithms it's the same, O(n^3).