Algorithmics	Student information	Date	Number of session
	UO: 300827	10/02/2025	1
	Surname: Leiras		
	Name: Sofía		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. currentTimeMillis()

Calculate how many more years we can continue using currentTimeMillis(). Explain what you did to calculate it.

Taking into account that the method currentTimeMillis() returns a long, which is a 64-bit signed integer. However, as we are taking about years, we have to take into account only the positive values, which are 2^{63} - 1. What is the same, we can use the method currentTimeMillis() to measure 2^{63} - 1 = 9,223,372,036,854,775,807 different milliseconds.

$$9{,}223{,}372{,}036{,}854{,}775{,}807 \text{ ms} \cdot \frac{1\ s}{1000\ ms} \cdot \frac{1\ min}{60\ s} \cdot \frac{1\ h}{60\ min} \cdot \frac{1\ day}{24\ h} \cdot \frac{1\ year}{365\ days}$$

$$\approx 292{,}471{,}208\ years$$

As the method started in 1970, we have to subtract the years the method has been already used in order to obtain the remaining ones:

$$Remaining\ years = 292{,}471{,}208 - (2025 - 1970) = 292{,}471{,}153\ years$$

We can continue using currentTimeMillis() for 292,471,153 more years before the value overflows.

Activity 2.

Why does the measured time sometimes come out as 0?

The most obvious reason that can explain this fact is that the time measured is lower than 1 milliseconds. What is the same, the difference between the time when the method currentTimeMillis() was first invoked and the time when it was secondly invoked is less than 1 milliseconds. In this case, to obtain the real time we would have to use another method which uses a smaller unit of time, so that the measurements can be precise enough to measure the difference.

However, there are other factors that can influence that time causing that it comes out as 0, such as the operating system. It can influence the measured execution time of a program by introducing interrupts, context switching, background processes… While running the

program, the CPU can switch to another process with higher level priority, such as the garbage collector run by the Java Virtual Machine, and all of this factors may alter the measured execution time.

From what size of problem (n) do we start to get reliable times?

SIZE=100000000 TIME=51 milliseconds SUM=-1515636

Times below 50 milliseconds are not taken into account because they are not considered reliable, so the first value of n with which I have obtained a time greater that 50ms when running Vector2.java is 10^8.

Activity 3. Taking small execution times

What happens with time if the problem size is multiplied by 2?

SIZE=10 TIME=40 milliseconds SUM=-154 NTIMES=10000000

SIZE=20 TIME=81 milliseconds SUM=3 NTIMES=10000000

SIZE=40 TIME=130 milliseconds SUM=201 NTIMES=10000000

SIZE=80 TIME=167 milliseconds SUM=-273 NTIMES=10000000

SIZE=160 TIME=415 milliseconds SUM=-488 NTIMES=10000000

SIZE=320 TIME=923 milliseconds SUM=-1549 NTIMES=10000000

SIZE=640 TIME=1779 milliseconds SUM=-260 NTIMES=10000000

SIZE=1280 TIME=3486 milliseconds SUM=-3434 NTIMES=10000000

With these results we can see that the execution time is more or less multiplied by two each time the size is multiplied by two.

What happens with time if the problem size is multiplied by a value k other than 2? (try it, for example, for k=3 and k=4 and check the times obtained)

Times when the problem size is multiplied by 3:

SIZE=10 TIME=42 milliseconds SUM=-333 NTIMES=10000000

SIZE=30 TIME=116 milliseconds SUM=264 NTIMES=10000000

SIZE=90 TIME=222 milliseconds SUM=193 NTIMES=10000000

SIZE=270 TIME=767 milliseconds SUM=-1515 NTIMES=10000000

Algorithmics	Student information	Date	Number of session
	UO: 300827	10/02/2025	1
	Surname: Leiras		
	Name: Sofía		

SIZE=810 TIME=2503 milliseconds SUM=1617 NTIMES=10000000

SIZE=2430 TIME=7682 milliseconds SUM=1701 NTIMES=10000000

SIZE=7290 TIME=23294 milliseconds SUM=4322 NTIMES=10000000

SIZE=21870 TIME=70810 milliseconds SUM=4744 NTIMES=10000000

Times when the problem size is multiplied by 4:

SIZE=10 TIME=61 milliseconds SUM=-118 NTIMES=10000000

SIZE=40 TIME=142 milliseconds SUM=-22 NTIMES=10000000

SIZE=160 TIME=431 milliseconds SUM=-1808 NTIMES=10000000

SIZE=640 TIME=2168 milliseconds SUM=1592 NTIMES=10000000

SIZE=2560 TIME=8921 milliseconds SUM=-2647 NTIMES=10000000

SIZE=10240 TIME=35244 milliseconds SUM=2814 NTIMES=10000000

Explain whether the times obtained are those expected from the linear complexity O(n)

Concluding from the previous results, we can assure that the linear complexity is true, because each time the size is multiplied by a factor, the execution time is also multiplied by the same factor.

TABLES (times in milliseconds WITHOUT OPTIMIZATION):

n	Tsum	Tmaximum	Tmatches1	Tmatches2
10000	$3.42 \cdot 10^{-3}$	$2.67 \cdot 10^{-3}$	35	$3.55 \cdot 10^{-3}$
20000	$6.33 \cdot 10^{-3}$	$5.58 \cdot 10^{-3}$	137	$7.05 \cdot 10^{-3}$
40000	$1.306 \cdot 10^{-2}$	$1.126 \cdot 10^{-2}$	439	$1.419 \cdot 10^{-2}$
80000	$2.817 \cdot 10^{-2}$	$2.190 \cdot 10^{-2}$	2102	$2.859 \cdot 10^{-2}$
160000	$5.278 \cdot 10^{-2}$	$4.412 \cdot 10^{-2}$	8474	$5.896 \cdot 10^{-2}$
320000	0.10642	$9.190 \cdot 10^{-2}$	33396	0.1666
640000	0.21074	0.18645	OoT	0.33922
1280000	0.42719	0.37857	OoT	0.776
2560000	1.150	0.864	OoT	1.871
5120000	2.786	2.650	OoT	3.529
10240000	5.392	5.125	OoT	7.102

Algorithmics	Student information	Date	Number of session
	UO: 300827	10/02/2025	1
	Surname: Leiras		
	Name: Sofía		

20480000	9.934	9.855	OoT	13.771
40960000	20.085	19.917	OoT	27.487
81920000	39.839	39.974	OoT	54.706

CPU: 11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz

RAM: 16,0 GB

Conclusions:

Tsum: the results obtained match the expected ones because the program has a linear complexity and the times are more or less multiplied by two (as well as the size):

Tmaximum: the same case as Tsum, the results are the expected ones.

Tmatches1: the results are the expected ones because the complexity of the program is quadratic and the times are also growing in a quadratic way.

Tmatches2: it is the same case as the first two and the results are the expected ones.