Algorithmics	Student information	Date	Number of session
	UO: 293615		
	Surname: Lavelle		
	Name: Gersan		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. How many more years can we continue using currentTimeMillis()?

The maximum amount of milliseconds we can represent is 9223372036854775807. That is equal to $2^{64} - 1$. The current value of the method on Monday 10th February 2025 at approximately 18:35 is 1739208934332. If we subtract both values we will get 18.44674233 * 10^{18}. That is the value of milliseconds of the time remaining until it reaches the maximum number. If we operate to convert the number in years we get 584942362.06240487062. So we can continue to use the method for just under 585 million years.

Activity 2. Measuring vector 2

The result will be 0 when the operation is executed so fast that it can't be measured properly.

I kept getting results of 0 milliseconds up until I tried with size 500000, which gave me a result of 2 milliseconds. This is still considered an unreliable measurement, so I kept trying with higher sizes, by adding a zero to the end of the number each time. When I got to size 50000000, I got 196 milliseconds, which surpasses the 50 milliseconds that was assigned as a reference of reliability.

Activity 3. Repetitions

If the size of the problem was multiplied by 2 instead of 5, the only thing that would change is the number of measurements that we will make, because the number of repetitions is applied in every iteration of the main for loop.

The lower number we use for the problem size multiplication, the higher the number of unreliable measurements we will obtain. Also, the higher we multiply the size by, the earlier we will get to the measurements that take a long time.

Algorithmics	Student information	Date	Number of session
	UO: 293615		
	Surname: Lavelle		
	Name: Gersan		

The time grows more or less proportional to the size, so the results obtained with linear complexity are the expected ones.

NTIMES Tsum: 20000.

NTIMES Tmaximum: 20000.

n	Tsum	Tmaximum
10000	69	70
20000	117	121
40000	233	231
80000	473	482
160000	950	950
320000	1918	1932
640000	3820	3799
1280000	7652	7366
2560000	16034	16148
5120000	34365	39092
10240000	OoT	OoT
20480000	OoT	OoT
40960000	OoT	OoT
81920000	OoT	OoT

Algorithmics	Student information	Date	Number of session
	UO: 293615		
	Surname: Lavelle		
	Name: Gersan		

NTIMES Tmatches1: 5

NTIMES Tmatches2: 20000

n	Tmatches1	Tmatches2
10000	170	66
20000	602	158
40000	2656	378
80000	7663	652
160000	29366	1393
320000	OoT	2845
640000	OoT	5733
1280000	OoT	12005
2560000	OoT	26030
5120000	OoT	53244
10240000	OoT	OoT
20480000	OoT	OoT
40960000	OoT	OoT
81920000	OoT	OoT

If I were to do matches 1 with the same number of repetitions that I used for matches 2, the execution time will be too long, and all values will we out of time.

Matches 2 is much faster because is has a linear complexity, while matches 2 has quadratic complexity.