Algorithmics	Student information	Date	Number of session
	UO: 301879	06/02/2025	1.1
	Surname: Sariego Sánchez		
	Name: Martín		

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 this way of counting.

Explain what you did to calculate it.

Type long holds 64 bits, so 2^64 -1 is the biggest number it can hold. This means that 18446744073709551615 milliseconds is the maximum time it can hold. This converts to approximately 584942417.355 years, and since it has been just over 55 years since January 1st 1970, we will be able to use this format for another 584942364.355 years.

Activity 2. Reliable times

Why does the measured time sometimes come out as 0?

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

The measured time comes out at zero sometimes because the vector is small, thus the sum of all the elements in the vector is done immediately.

At n= 13000000 we get t= 50 ms, so n values after that will give reliable times > 50 ms.

Activity 3. Taking small execution times

If problem size is multiplied by two, due to the sum complexity being O(n), the time will also be multiplied by two. If the problem size was multiplied by 3 or 4, or any other number, execution time would also be multiplied by the same number, since it's linear complexity.

C1(lab)	
CPU	i5-12400
RAM	16,0 GB

Algorithmics	Student information	Date	Number of session
	UO: 301879	06/02/2025	1.1
	Surname: Sariego Sánchez		
	Name: Martín		

Table 1:

n	Tsum(ms)	Tmaximum
10000	0,039	0,056
20000	0,074	0,112
40000	0,149	0,218
80000	0,296	0,447
160000	0,593	0,882
320000	1,182	1,757
640000	2,362	3,516
1280000	4,732	7,051
2560000	9,536	14,214
5120000	19,278	29,437
10240000	39,143	58,103
20480000	78,462	116,374
40960000	156,236	231,181
81920000	313,142	458,948

Table 2:

n	Tmatches1	Tmatches2
10000	509	0,06
20000	2023	0,116
40000	8110	0,338
80000	32296	0,468
160000	129225	0,929
320000	Oot	1,852
640000	Oot	3,615
1280000	Oot	7,245
2560000	Oot	14,761
5120000	Oot	29,694
10240000	Oot	58,364
20480000	Oot	119,038
40960000	Oot	238,385
81920000	Oot	473,204

All of these algorithms are O(n) complexity, except matches1, which is O(n^2), so it makes sense that the times increase by k=2 (k being the increase in problem size), and that in matches1, the time increases by k^2=4, as expected.