PABLO GARCÍA FERNÁNDEZ UO276824

Activity 1. Measuring execution times

1. How many more years can we continue using this way of counting?

   We can use it until it overflows that is in 292 million years.

2. What does it mean that the time measured is 0?

   It means that it takes less than a millisecond to the program to be executed.

3. From what size of the problem do we start to get reliable times?

   We start getting reliable times at n = 1000000

Activity 2. Grow of the problem size

1. What happens with time if the size of the problem is multiplied by 5?

   Happens that Vector2 takes less time than Vector3 to execute the method because the size of Vector3 increases exponentially if in both classes you introduce the same size.

2. Are the times obtained those that were expected from linear complexity O(n)?

   Yes, because it takes a little and it increases in a linear way.

3. Use a spreadsheet to draw a graph with Excel. On the X axis we can put the time and on the Y axis the size of the problem.

N	Vector2(ms)	Vector3(ms)

1	0	0
10	0	0
100	0	0
1000	0	0
10000	0	1
100000	1	3
1000000	4	6
10000000	7	25
100000000	43	197

Activity 3. Taking small execution times

n	fillIn(t)*10^-7 ms	sum(t)*10^-7 ms	maximum(t)*10^-7 ms
10	32	100	1296
30	56	128	1765
90	209	352	7130
270	759	972	20596
810	2520	2710	60630
2430	7670	7856	182428
7290	22900	23496	541520
21870	68912	69588	1622015
65610	More than an hour	More than an hour	More than an hour
196830	More than an hour	More than an hour	More than an hour
590490	More than an hour	More than an hour	More than an hour
1771470	More than an hour	More than an hour	More than an hour

1. What are the main components of the computer in which you did the work (process, memory)?

   AMD Ryzen7 1700 Eight-Cores 3.00 GHz
   16 GB RAM DDR4

Activity 4. Operations on matrices

n	sumDiagonal1(t)*10^-6 ms	sumDiagonal2(t)*10^-6 ms
10	79	23
30	503	34
90	4609	78

270	38483	312
810	324108	1189
2430	2773059	27403
7290	Crash	121930
21870	Crash	Crash
65610	Crash	Crash
196830	Crash	Crash
590490	Crash	Crash
1771470	Crash	Crash

1. What are the main components of the computer in which you did the work (process, memory)?

   AMD Ryzen7 1700 Eight-Cores 3.00 GHz
   16 GB RAM DDR4

2. Do the values obtained meet the expectations?

   SumDiagonal1() is O(n^2) and it looks like it in the practical measures and SumDiagonal2() is O(n) and the measures look like it.

Activity 5. Benchmarking

1. Why you get differences in execution time between the two programs?

   Because of the different efficiency in both languages.

2. Regardless of the specific times, is there any analogy in the behaviour of the two implementations?

   Also different implementation as it is done changes performance because none of both languages are equal so their implementation is different so the execution time it is also different.