	Student information	Date	Number of session
Algorithmics	UO: 294039	08/02/2024	1.1
	Surname: Rodriguez Gomara		
	Name: Maria		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. Measuring execution times

Calculate how many more years we can continue using this way of counting.

We can continue using this way of counting for 292276970 years.

What does it mean that the time measured is 0?

It can means two thing:

1. The execution time was small
2. The difference between both time stamps is the same as the time your operating system needs to resolute time

If we want to get more precise results we could use the method System.nanoTime()

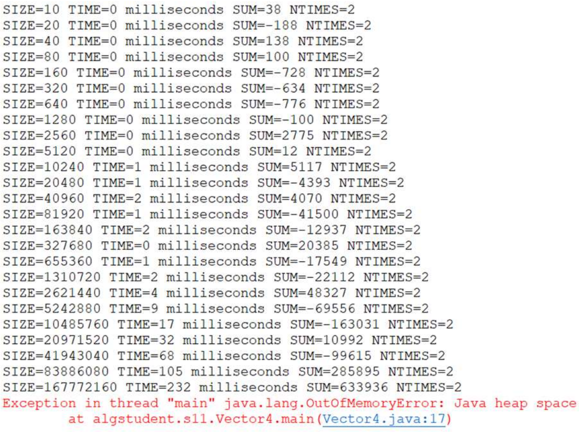
From what size of problem (n) do we start to get reliable times?

With size 150000000 we obtain an execution time of 56 milliseconds.

Activity 2. Taking small execution times

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

The execution time increases linearly.

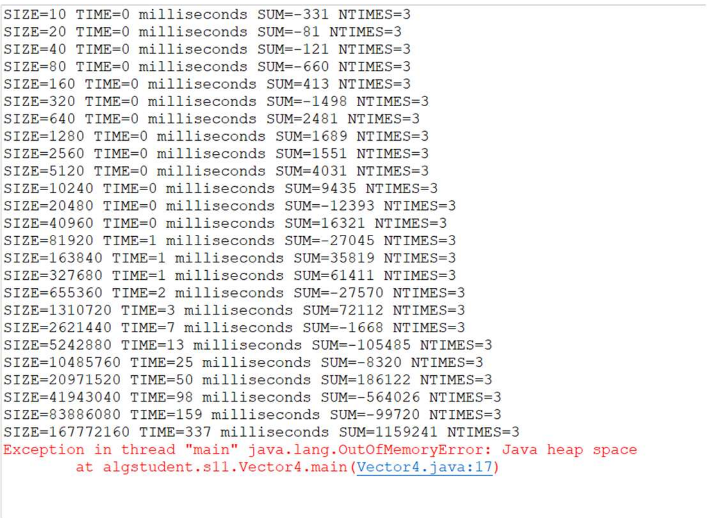


```
SIZE=10 TIME=0 milliseconds SUM=38 NTIMES=2
SIZE=20 TIME=0 milliseconds SUM=-188 NTIMES=2
SIZE=40 TIME=0 milliseconds SUM=138 NTIMES=2
SIZE=80 TIME=0 milliseconds SUM=100 NTIMES=2
SIZE=160 TIME=0 milliseconds SUM=-728 NTIMES=2
SIZE=320 TIME=0 milliseconds SUM=-634 NTIMES=2
SIZE=640 TIME=0 milliseconds SUM=-776 NTIMES=2
SIZE=1280 TIME=0 milliseconds SUM=-100 NTIMES=2
SIZE=2560 TIME=0 milliseconds SUM=2775 NTIMES=2
SIZE=5120 TIME=0 milliseconds SUM=12 NTIMES=2
SIZE=10240 TIME=1 milliseconds SUM=5117 NTIMES=2
SIZE=20480 TIME=1 milliseconds SUM=-4393 NTIMES=2
SIZE=40960 TIME=2 milliseconds SUM=4070 NTIMES=2
SIZE=81920 TIME=1 milliseconds SUM=-41500 NTIMES=2
SIZE=163840 TIME=2 milliseconds SUM=-12937 NTIMES=2
SIZE=327680 TIME=0 milliseconds SUM=20385 NTIMES=2
SIZE=655360 TIME=1 milliseconds SUM=-17549 NTIMES=2
SIZE=1310720 TIME=2 milliseconds SUM=-22112 NTIMES=2
SIZE=2621440 TIME=4 milliseconds SUM=48327 NTIMES=2
SIZE=5242880 TIME=9 milliseconds SUM=-69556 NTIMES=2
SIZE=10485760 TIME=17 milliseconds SUM=-163031 NTIMES=2
SIZE=20971520 TIME=32 milliseconds SUM=10992 NTIMES=2
SIZE=41943040 TIME=68 milliseconds SUM=-99615 NTIMES=2
SIZE=83886080 TIME=105 milliseconds SUM=285895 NTIMES=2
SIZE=167772160 TIME=232 milliseconds SUM=633936 NTIMES=2
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
	at algstudent.s11.Vector4.main(Vector4.java:17)
```

What happens with the 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)

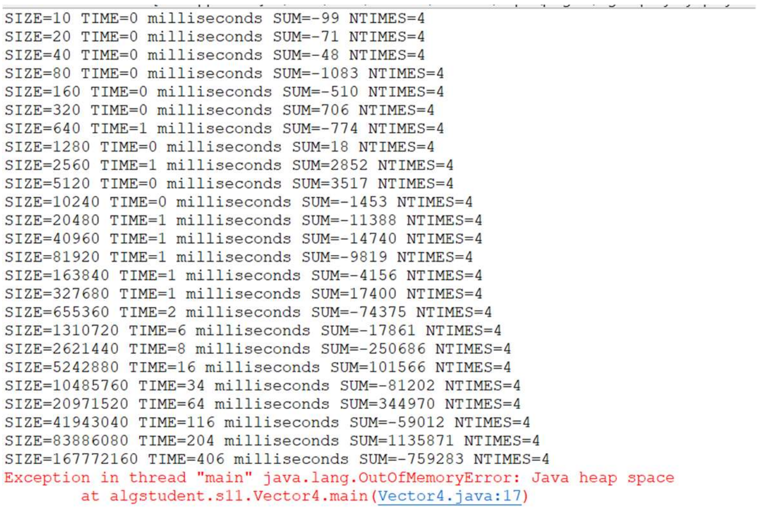
k = 3

Algorithmics	Student information	Date	Number of session
	UO: 294039	08/02/2024	1.1
	Surname: Rodriguez Gomara		
	Name: Maria		



```
SIZE=10 TIME=0 milliseconds SUM=-331 NTIMES=3
SIZE=20 TIME=0 milliseconds SUM=-81 NTIMES=3
SIZE=40 TIME=0 milliseconds SUM=-121 NTIMES=3
SIZE=80 TIME=0 milliseconds SUM=-660 NTIMES=3
SIZE=160 TIME=0 milliseconds SUM=413 NTIMES=3
SIZE=320 TIME=0 milliseconds SUM=-1498 NTIMES=3
SIZE=640 TIME=0 milliseconds SUM=2481 NTIMES=3
SIZE=1280 TIME=0 milliseconds SUM=1689 NTIMES=3
SIZE=2560 TIME=0 milliseconds SUM=1551 NTIMES=3
SIZE=5120 TIME=0 milliseconds SUM=4031 NTIMES=3
SIZE=10240 TIME=0 milliseconds SUM=9435 NTIMES=3
SIZE=20480 TIME=0 milliseconds SUM=-12393 NTIMES=3
SIZE=40960 TIME=0 milliseconds SUM=16321 NTIMES=3
SIZE=81920 TIME=1 milliseconds SUM=-27045 NTIMES=3
SIZE=163840 TIME=1 milliseconds SUM=35819 NTIMES=3
SIZE=327680 TIME=1 milliseconds SUM=61411 NTIMES=3
SIZE=655360 TIME=2 milliseconds SUM=-27570 NTIMES=3
SIZE=1310720 TIME=3 milliseconds SUM=72112 NTIMES=3
SIZE=2621440 TIME=7 milliseconds SUM=-1668 NTIMES=3
SIZE=5242880 TIME=13 milliseconds SUM=-105485 NTIMES=3
SIZE=10485760 TIME=25 milliseconds SUM=-8320 NTIMES=3
SIZE=20971520 TIME=50 milliseconds SUM=186122 NTIMES=3
SIZE=41943040 TIME=98 milliseconds SUM=-564026 NTIMES=3
SIZE=83886080 TIME=159 milliseconds SUM=-99720 NTIMES=3
SIZE=167772160 TIME=337 milliseconds SUM=1159241 NTIMES=3
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
	at algstudent.s11.Vector4.main(Vector4.java:17)
```

K = 4



```
SIZE=10 TIME=0 milliseconds SUM=-99 NTIMES=4
SIZE=20 TIME=0 milliseconds SUM=-71 NTIMES=4
SIZE=40 TIME=0 milliseconds SUM=-48 NTIMES=4
SIZE=80 TIME=0 milliseconds SUM=-1083 NTIMES=4
SIZE=160 TIME=0 milliseconds SUM=-510 NTIMES=4
SIZE=320 TIME=0 milliseconds SUM=706 NTIMES=4
SIZE=640 TIME=1 milliseconds SUM=-774 NTIMES=4
SIZE=1280 TIME=0 milliseconds SUM=18 NTIMES=4
SIZE=2560 TIME=1 milliseconds SUM=2852 NTIMES=4
SIZE=5120 TIME=0 milliseconds SUM=3517 NTIMES=4
SIZE=10240 TIME=0 milliseconds SUM=-1453 NTIMES=4
SIZE=20480 TIME=1 milliseconds SUM=-11388 NTIMES=4
SIZE=40960 TIME=1 milliseconds SUM=-14740 NTIMES=4
SIZE=81920 TIME=1 milliseconds SUM=-9819 NTIMES=4
SIZE=163840 TIME=1 milliseconds SUM=-4156 NTIMES=4
SIZE=327680 TIME=1 milliseconds SUM=17400 NTIMES=4
SIZE=655360 TIME=2 milliseconds SUM=-74375 NTIMES=4
SIZE=1310720 TIME=6 milliseconds SUM=-17861 NTIMES=4
SIZE=2621440 TIME=8 milliseconds SUM=-250686 NTIMES=4
SIZE=5242880 TIME=16 milliseconds SUM=101566 NTIMES=4
SIZE=10485760 TIME=34 milliseconds SUM=-81202 NTIMES=4
SIZE=20971520 TIME=64 milliseconds SUM=344970 NTIMES=4
SIZE=41943040 TIME=116 milliseconds SUM=-59012 NTIMES=4
SIZE=83886080 TIME=204 milliseconds SUM=1135871 NTIMES=4
SIZE=167772160 TIME=406 milliseconds SUM=-759283 NTIMES=4
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
	at algstudent.s11.Vector4.main(Vector4.java:17)
```

Every time you increase the problem size of Vector 4 the execution time increases linearly, as you can see on the screenshots.

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

With the times obtained from the previous classes (in milliseconds), fill in the following two tables:

n	Tsum(ms)	Tmaximum(ms)
10000	0,179	0,206
20000	0,336	0,400
40000	0,657	0,772

80000	1,337	1,595
160000	2,654	3,320
320000	5,980	7,310
640000	12,586	16,342
1280000	24,965	30,131
2560000	49,962	61,396
5120000	98,373	113,760
10240000	185,462	218,525
2048000	355,107	435,104
40960000	718,147	944,521
81920000	1395,177	1926,879

n	Tmatches1(ms)	Tmatches2(ms)
10000	1579	0,271
20000	6483	0,446
40000	27796	0,853
80000	117365	1,684
160000	463110	3,439
320000	1780411	6,702
640000	3998645	13,356
1280000	--	29,438
2560000	--	59,358
5120000	--	117,534
10240000	--	229,988
2048000	--	473,423
40960000	--	972,165

81920000	--	2570,883

The complexities of each functions are:

- Sum -> O(n)
- Maximum -> O(n)
- Matches1 -> O(n^2)
- Matches2 -> O(n)

The times met what was expected because Matches1 has a higher complexity than the other programs and that it is reflected in the execution time.

For this session I use a computer with:

Procesador -> AMD Ryzen 5 Microsoft Surface (R) Edition   2.20 GHz

RAM -> instalada 8,00 GB (7,45 GB usable)