Activity 1. Measuring execution times

How many years can we continue using System. currentTimeMillis()

The return type of this method is long, that means that we have 64bits to represent it.

264 – 1 = 9.223.372.036.854.775.807

This is the maximum number of ms we can represent. At the moment I’m writing this document, the return of System.currentTimeMillis() is 1.739.208.680.863.

Then by a simple calculation we can get the available ms we still have until this approach stops working: 9.223.372.036.854.775.807 - 1.739.208.680.863 = 1.844674233\*1019 ms

We have to convert these milliseconds into years, I have used [this web](https://www.inchcalculator.com/convert/millisecond-to-year/) ang got the following result: 584.553.993,9979 years

Vector 2

Our problem complexity is O(n), first it fills an array of a size passed as argument by the user and then adds all the elements in that array.

As it has a good complexity, we need a big problem size to get measuring times.

In my computer, I start getting reliable times with a problem size >= 15.000.000

Activity 2. Taking small execution times

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

With a number of 1.000.000 repetitions I got the following results:

|  |  |
| --- | --- |
| n | time(micros) |
| 10 | 88 |
| 50 | 270 |
| 250 | 1085 |
| 1250 | 5196 |
| 6250 | 26139 |
| 31250 | OoT |
| 156250 | OoT |
| 781250 | OoT |
| 3906250 | OoT |

If the problem size is multiplied by two:

|  |  |
| --- | --- |
| n | times(micros) |
| 10 | 81 |
| 20 | 123 |
| 40 | 211 |
| 80 | 366 |
| 160 | 701 |
| 320 | 1345 |
| 640 | 2598 |
| 1280 | 5157 |
| 2560 | 10490 |
| 5120 | 21621 |
| 10240 | 41763 |

Times follow the Linear complexity, everytime n grows as 2, time grows more or less by 2 also. For example, n=20 t= 123 // n=40 t=211 which is almost the double for the time.

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)

Increased by 3:

|  |  |
| --- | --- |
| n | time(micros) |
| 10 | 85 |
| 30 | 161 |
| 90 | 402 |
| 270 | 1109 |
| 810 | 3175 |
| 2430 | 9570 |
| 7290 | 28876 |
| 21870 | OoT |

Increased by 4:

|  |  |
| --- | --- |
| n | time(micros) |
| 10 | 82 |
| 40 | 200 |
| 160 | 679 |
| 640 | 2525 |
| 2560 | 10057 |
| 10240 | 40619 |
| 40960 | OoT |

We can see how the times grow linearly following the complexity.