Activity 1. currentTimesMillis()

The maximum long value for java, will correspond to 9,223,372,036,854,775,807 milliseconds from the 1st January, 1970. By factors of conversion, we can change ms to a more appropriate measure of time such as years.

As we used it for 55 years, we could use this system for 1.169.084.037 years more.

Activity 2. Reliable times

Sometimes the measured time comes out as 0 because the operation is too fast to be measured, or in other words, it takes less than a millisecond to perform it.

The time starts getting reliable around size 10,000,000, for which the time is 59 ms.

Activity 3. Vectors

When executing vector4, if the problem size is multiplied by 2, i.e., k = 2, the time it takes to complete the algorithm doubles as well. The times increase linearly proportional to the size, so if instead of multiplied by 2, the problem size gets multiplied by 3 or 4, the times do it as well.

These times increment makes according to the complexity of the algorithm used (O(n), which is linear).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | Tsum | Tmaximum | Tmatches1 | Tmatches2 |
| 10000 | 0.0605 | 0.0792 | 601 | 0.0783 |
| 20000 | 0.1202 | 0.1559 | 2328 | 0.1539 |
| 40000 | 0.2302 | 0.3025 | 9264 | 0.2975 |
| 80000 | 0.4762 | 0.5993 | 36675 | 0.5894 |
| 160000 | 0.9276 | 1.2272 | OoT | 1.1839 |
| 320000 | 1.8956 | 2.4435 | OoT | 2.3867 |
| 640000 | 3.785 | 4.8586 | OoT | 5.06 |
| 1280000 | 8.3 | 10.5 | OoT | 10.21 |
| 2560000 | 15.4 | 20.4 | OoT | 18.91 |
| 5120000 | 3139 | 39.2 | OoT | 38.04 |
| 10240000 | 61.7 | 83.1 | OoT | 76.17 |
| 20480000 | 130 | 156.7 | OoT | 151.46 |
| 40960000 | 255.3 | 325.1 | OoT | 303.8 |
| 81920000 | 491.1 | 623.9 | OoT | 607.43 |

Processor: Intel Core i5-11400

Memory RAM: 16GB

Their complexities are:

Tsum: O(n)

Tmaximum: O(n)

Tmatches1: O(n2)

Tmatches2: O(n)

The times match their complexity as the size is multiplied by 2 so the times double in all cases, except in matches1(), which gets multiplied by 22 = 4.