Lab task 1. Empirical study of the complexity of algorithms

HEXAGONAL NUMBERS

Problem description

A hexagonal number [1] is a polygonal number [2] that represents a hexagon. The hexagonal number for n is given by the formula,

$$H_n = n\,(2n - 1) = 2n^2 - n \tag{1}$$

However, it also can be obtained as the nth term of the series,

$$H_n = \sum_{i=0}^{n-1}(4i + 1) \tag{2}$$

You are asked to design three algorithms implemented in Java that, using n as input parameter, provide the nth hexagonal number. The first algorithm must use equation (1) and the other two equation (2) in iterative and recursive form. You should:

1) Determine theoretically the complexity of the algorithms
2) Determine empirically the running time of the algorithms as a function of n
   a. In this case, assign different values to n and, for each of them, calculate the running time of each method. To do this, you can use the `currentTimeMillis` and/or `nanoTime` methods of the `System` class, which provide the system time in milliseconds and nanoseconds, respectively.

Each class and method in the program should include appropriate internal documentation to make it fully understandable. The InternalDocumentation.pdf file (lab section in *Campus Virtual*) contains a brief guide to writing internal documentation.

References

1. https://en.wikipedia.org/wiki/Hexagonal_number

2. https://en.wikipedia.org/wiki/Polygonal_number