Task 1. Experimental time complexity analysis

Goal

Experimental study of the time complexity of different algorithms

Problems and methods

For each n from 1 to 2000, measure the average computer execution time (using timestamps) of programs implementing the algorithms and functions below for five runs. Plot the data obtained showing the average execution time as a function of n. Conduct the theoretical analysis of the time complexity of the algorithms in question and compare the empirical and theoretical time complexities.

I. *Generate an n-dimensional random vector* $\boldsymbol{v} = [v_1, v_2, \dots, v_n]$ *with non-negative elements. For* $\boldsymbol{v}$*, implement the following calculations and algorithms:*

1) $f(\boldsymbol{v}) = const$ *(constant function);*
2) $f(\boldsymbol{v}) = \sum_{k=1}^{n} v_k$ *(the sum of elements);*
3) $f(\boldsymbol{v}) = \prod_{k=1}^{n} v_k$ *(the product of elements);*
4) *supposing that the elements of* $\boldsymbol{v}$ *are the coefficients of a polynomial* P *of degree* $n-1$*, calculate the value* $P(1.5)$ *by a direct calculation of* $P(x) = \sum_{k=1}^{n} v_k x^{k-1}$ *(i.e. evaluating each term one by one) and by Horner's method by representing the polynomial as* $P(x) = v_1 + x\big(v_2 + x(v_3 + \cdots)\big)$*;*
5) *Bubble Sort of the elements of* $\boldsymbol{v}$*;*
6) *Quick Sort of the elements of* $\boldsymbol{v}$*;*
7) *Timsort of the elements of* $\boldsymbol{v}$*.*

II. *Generate random matrices* A *and* B *of size* $n \times n$ *with non-negative elements. Find the usual matrix product for* A *and* B*.*

III. *Describe the data structures and design techniques used within the algorithms.*