FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION

OF HIGHER EDUCATION

ITMO UNIVERSITY

Report

on the practical task No. 1

“Experimental time complexity analysis”

Performed by

*Konstantin Krechetov*

*Academic group: j4132c*

Accepted by

Dr Petr Chunaev

St. Petersburg

2020

**Goal**

*Experimental study of the time complexity of different algorithms*

**Formulation of the problem**

*Considering some methods such as sorting, multiplying matrices and the calculation of the polynomial. Each theoretical estimations should be obtained and compared to empirical ones.*

**Brief theoretical part**

*All problems can be solved with more than one algorithm. Each of them may stand for different methodological approaches and take the appropriate time. For example, bubble sort is O(n^2) algorithm while quicksort is better because it takes O(nlog(n)) time on average.*

*Time complexity is measured in terms of O(n) because all CPUs have various computational power. The same program code may take different time on two computers but the common ‘scale’ to measure it is the number of computer operations.*

*There are few complex approaches to solve problems: divide and conquer, greedy algorithms, naïve ones. For example, merge sort is one of ‘divide and conquer’ algorithms (but it is not considered in this work).*

**Results**

*Theoretical and empirical estimation for each method are demonstrated on figures below. Theoretical time complexity:*

*Quicksort = O(nlog(n)) average, O(n2) worst. I approximate it with O(nlog(n)) because each time I create* ***random*** *vector.*

*f\_const: O(const)*

*f\_sum: O(n)*

*f\_prod: O(n)*

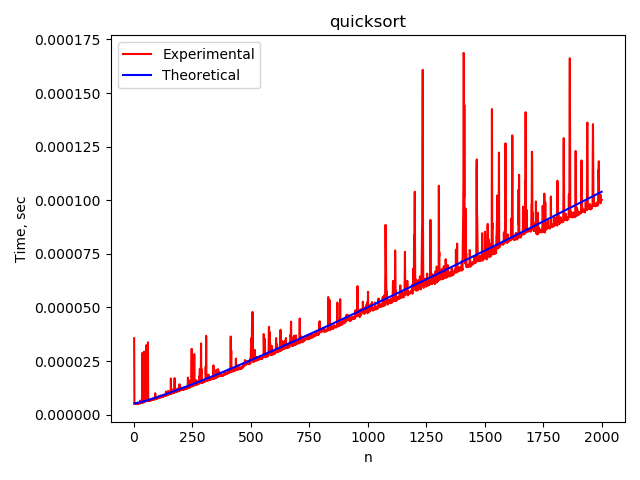
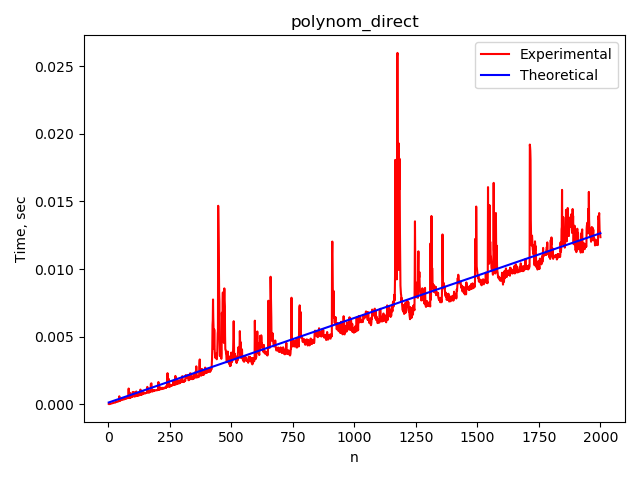
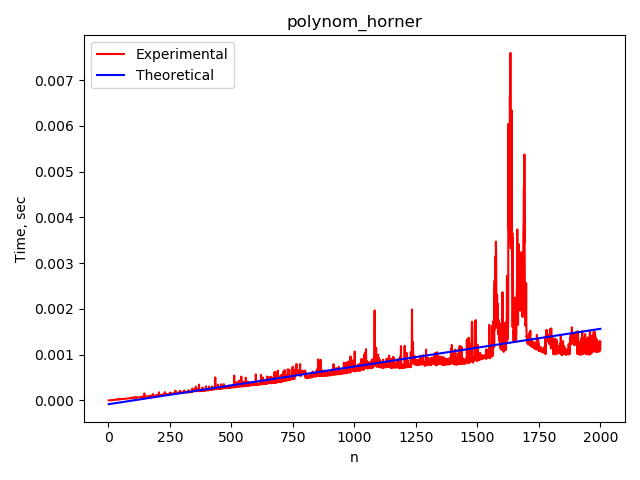
*polynom\_direct: O(n)*

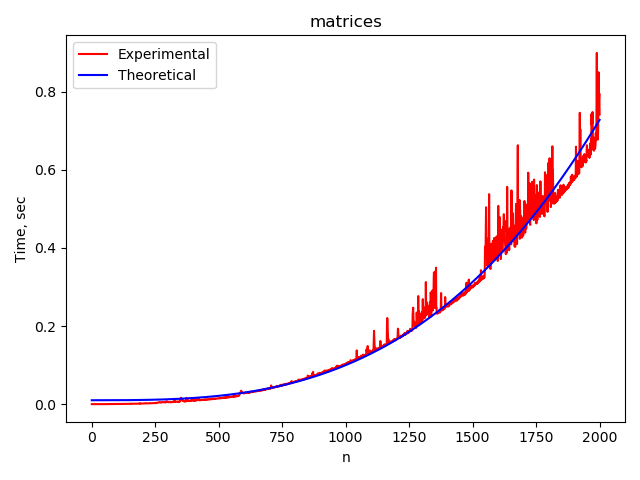
*polynom\_horner: O(n)*

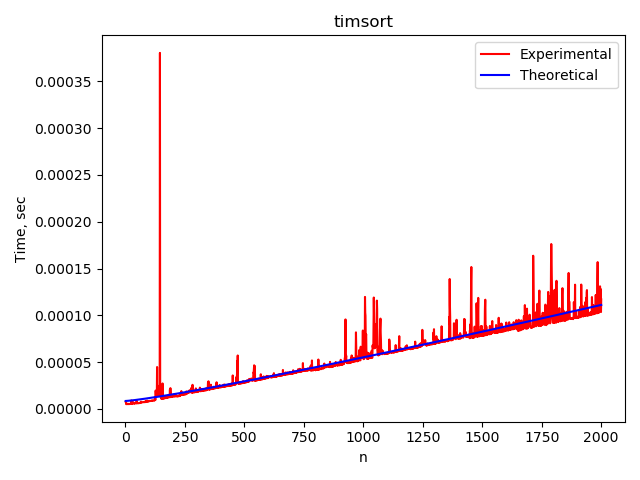
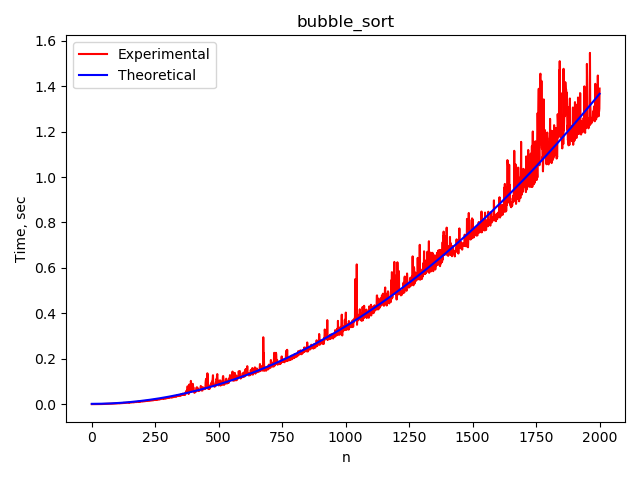
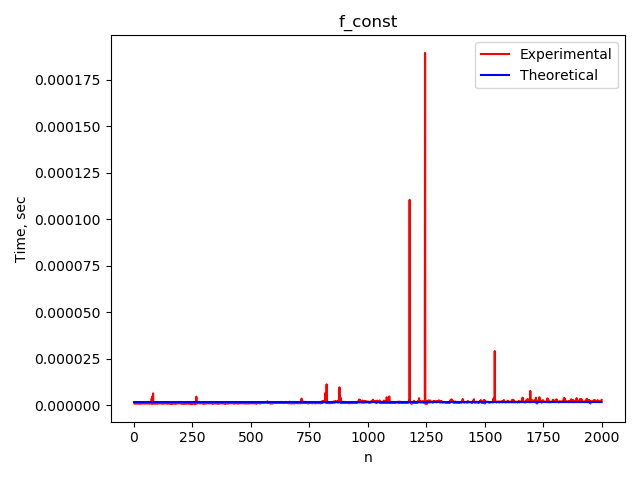
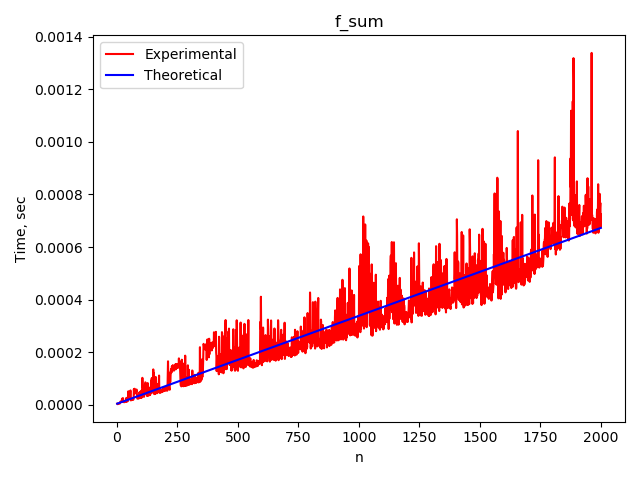
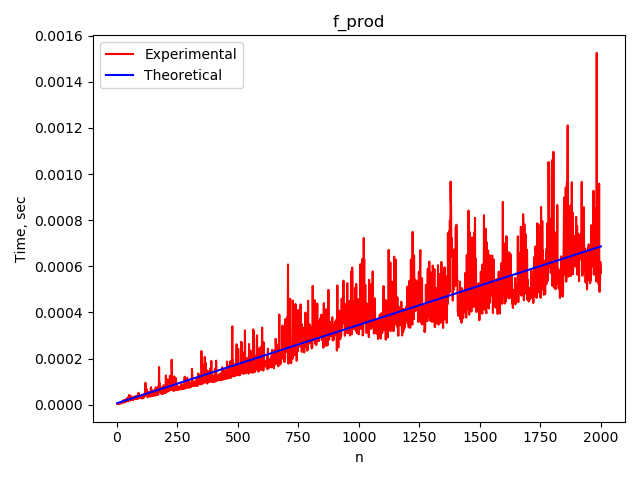
*bubble\_sort: O(n2)*

*timsort: O(nlogn)*

*matrices: O(n3)*







**Conclusions**

*Each empirical execution time matches its theoretical one. There are some fluctuations over the latter curve but the trend in total fits good.*

*This work introduces to algorithms complexity, its approaches and demonstrates the importance of using appropriate algorithm. As we can see on the plots, having O(nlogn) algorithm has a great benefit against O(n^2) for n >> 1.*

**Appendix**

*Source code available on*

<https://github.com/KostyaKrechetov/ITMO-Analysis-and-development-of-algorithms/tree/master/Task1>