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(n^2) 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)

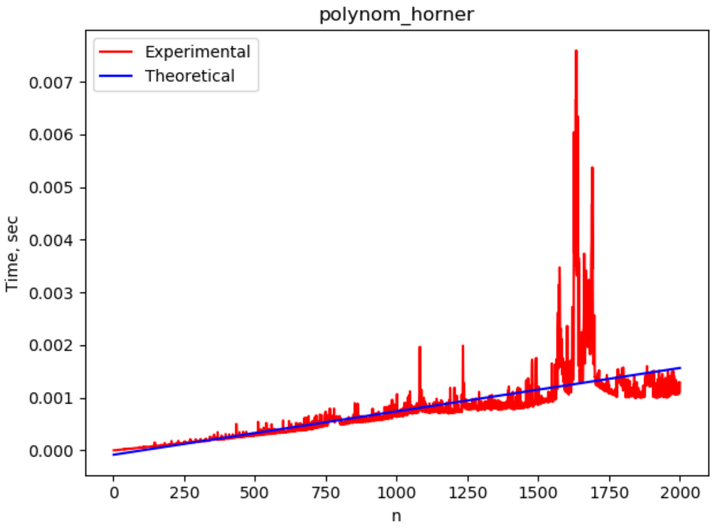
polynom_direct: O(n)

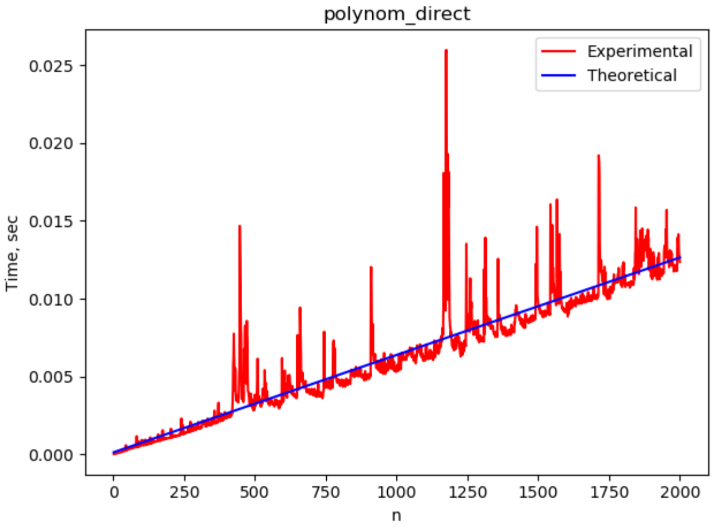
polynom_horner: O(n)

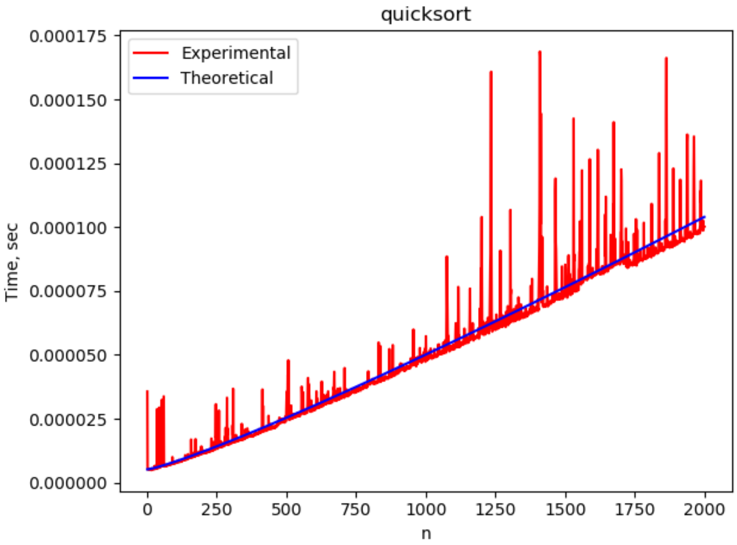
bubble_sort: O(n^2)

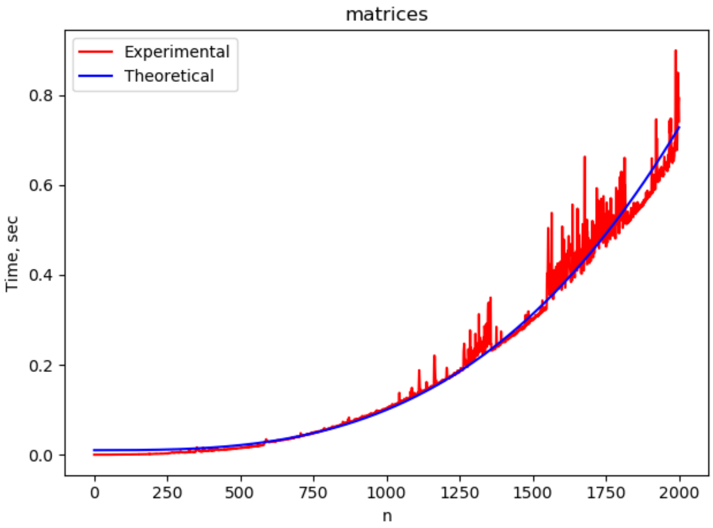
timsort: O(nlogn)

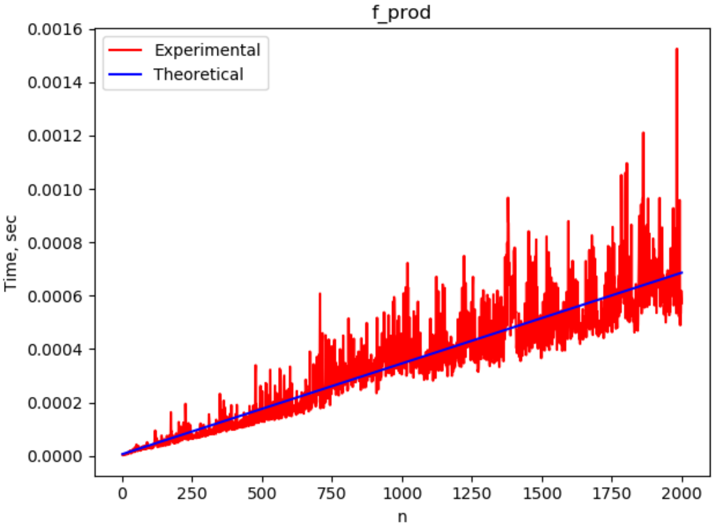
matrices: O(n^3)

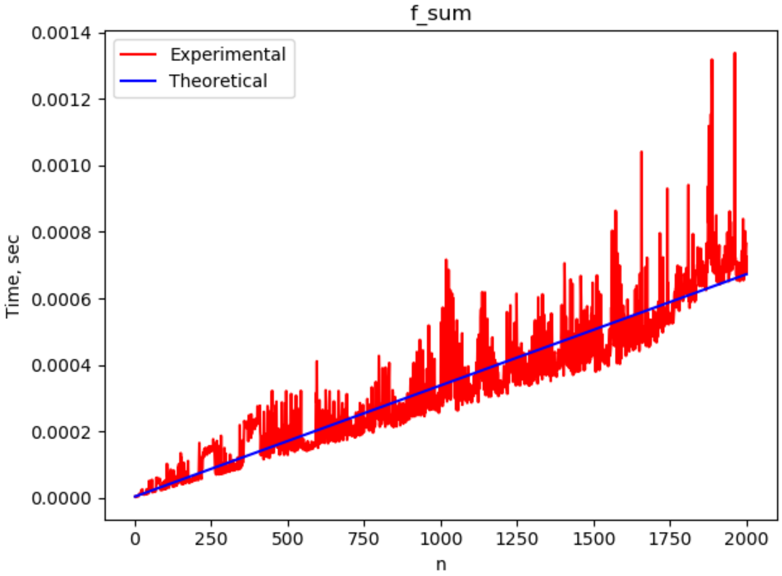

polynom_horner
Experimental
Theoretical
Time, sec
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0
250
500
750
1000
1250
1500
1750
2000
n

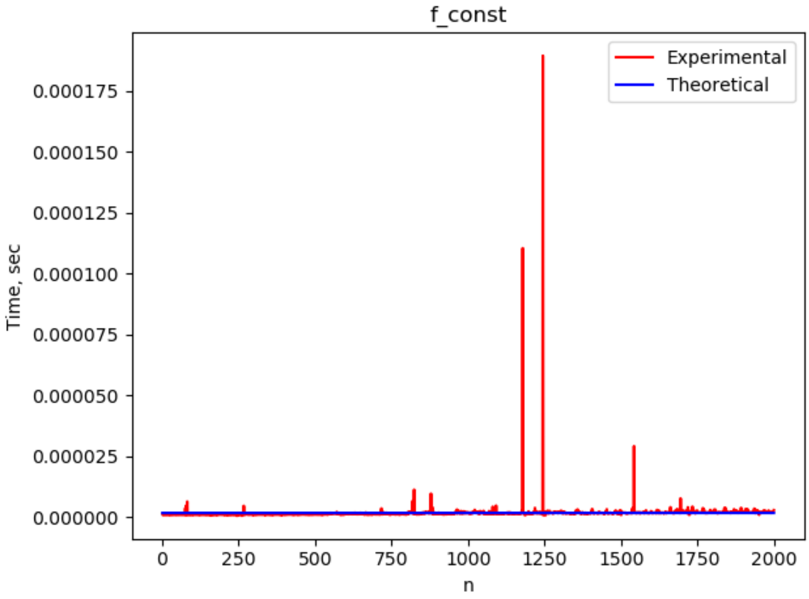

polynom_direct
Experimental
Theoretical
Time, sec
0.000
0.005
0.010
0.015
0.020
0.025
0
250
500
750
1000
1250
1500
1750
2000
n

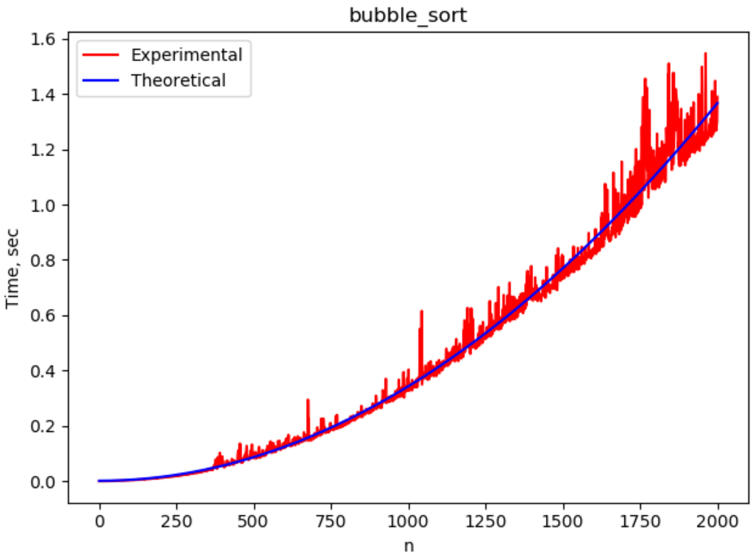

quicksort
Experimental
Theoretical
Time, sec
0.000000
0.000025
0.000050
0.000075
0.000100
0.000125
0.000150
0.000175
0
250
500
750
1000
1250
1500
1750
2000
n

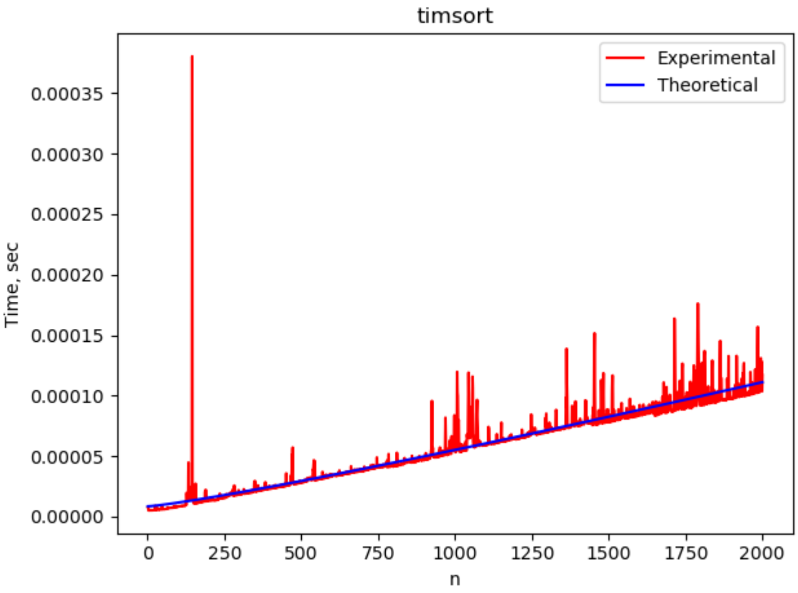

matrices
Experimental
Theoretical
Time, sec
0.0
0.2
0.4
0.6
0.8
0
250
500
750
1000
1250
1500
1750
2000
n


f_prod
Experimental
Theoretical
Time, sec
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
0.0012
0.0014
0.0016
0
250
500
750
1000
1250
1500
1750
2000
n


f_sum
Experimental
Theoretical
Time, sec
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
0.0012
0.0014
0
250
500
750
1000
1250
1500
1750
2000
n


f_const
Experimental
Theoretical
Time, sec
0.000175
0.000150
0.000125
0.000100
0.000075
0.000050
0.000025
0.000000
0
250
500
750
1000
1250
1500
1750
2000
n


bubble_sort
Experimental
Theoretical
Time, sec
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
250
500
750
1000
1250
1500
1750
2000
n


timsort
Experimental
Theoretical
Time, sec
0.00035
0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
0
250
500
750
1000
1250
1500
1750
2000
n

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