CS201
Fundamental Structures of Computer Science I

Bilkent University
Spring 2021/2022

Homework Assignment 2
06.04.2022, Wednesday

Section 1

Deniz Tuna Onguner
22001788

Algorithm Analysis

Algorithm 1

This algorithm takes base(a), power(n), and modulus(p) as parameters and in a single for loop it calculates the $a^n \pmod p$. The for loop always executes n number of times before returns the result, so that the execution speed of the algorithm is only dependent on variable n. Other variables, a and p, do not alter the speed. Consequently, the big O notation of this algorithm is **O(n)**.

Algorithm 2

Similar to the very first algorithm, this function's worst case big O notation is also **O(n)**. Because in the worst case situation, the function's behavior is exactly same with the first one. However, as seen from the table and graphs below, besides the worst case scenario, with better n values, the algorithm runs significantly faster compared to the first one. Because with such such values, result returns with less loop executions. In case of the values from the table below, the function runs like its speed is O(1) since that case is not a worst one.

Algorithm 3

Unlike first two functions, algorithm three has no loop at all, but it is called recursively. The problem, $a^n \pmod p$, is split into smaller problems in each step, so that the function returns faster. It can be calculated that the big O notation is **O(log n)**. Since there is no special cases for this function, which is an important different with the second one, this one is the most efficient one compared to the others. Even though the second might return faster with specifically selected parameters, still, this one also returns quickly even with the values that are the worst case of the second one.

Device Specifications

MacBook Pro (16-inch, 2019), Apple

CPU[1]:

Intel® Core™ i7-9850H Processor (9th Generation)

Total Cores: 6

Total Threads: 12

Cache: 12 MB

Max Turbo Frequency: 4.60 GHz

Processor Base Frequency: 2.60 GHz

Memory:

2 x 8 GB/ DDR4/ 2667 MHz,

16 GB in total

Operating System:

macOS Monterey,

version 12.2.1

Development Environment:

IDE: CLion 2021.3.4, JetBrains

Compiler: C++98

[1] *Product specifications*. Intel Core i79850H Processor 12M Cache up to 4.60 GHz Product Specifications. (n.d.). Retrieved March 31, 2022, from https://ark.intel.com/content/www/us/en/ark/products/191047/intel-core-i79850h-processor-12m-cache-up-to-4-60-ghz.html

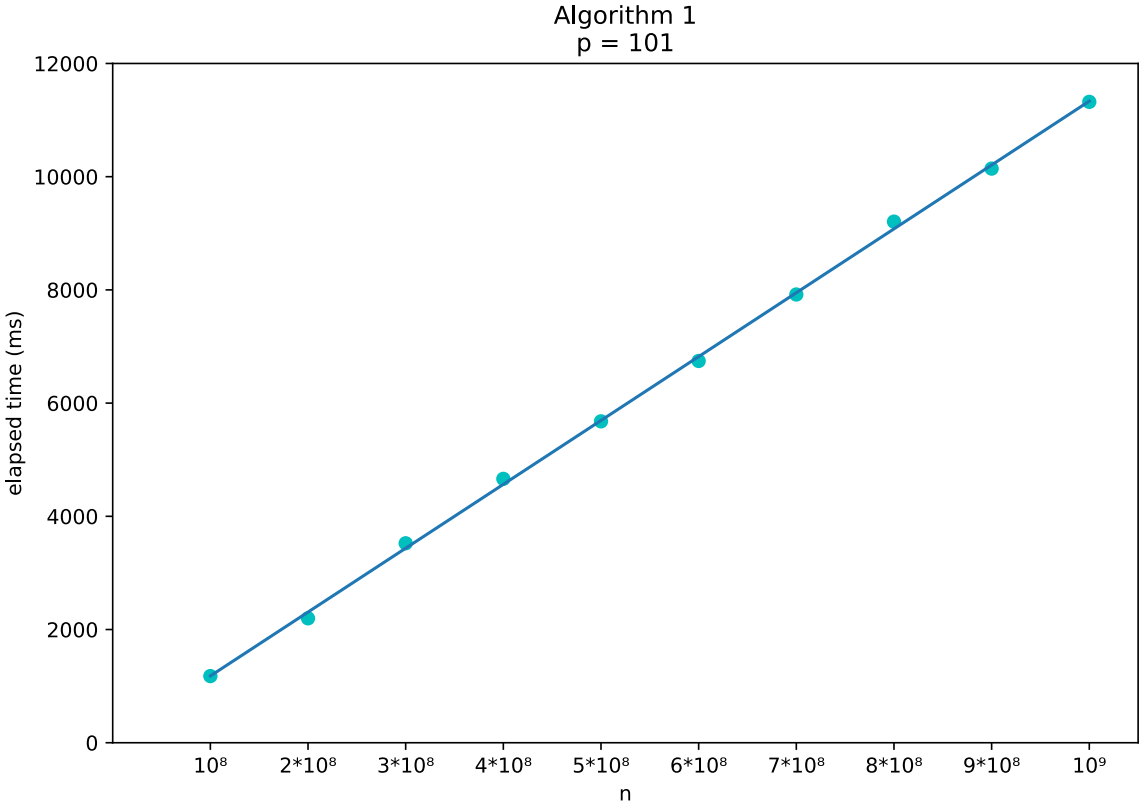
Execution Time Results

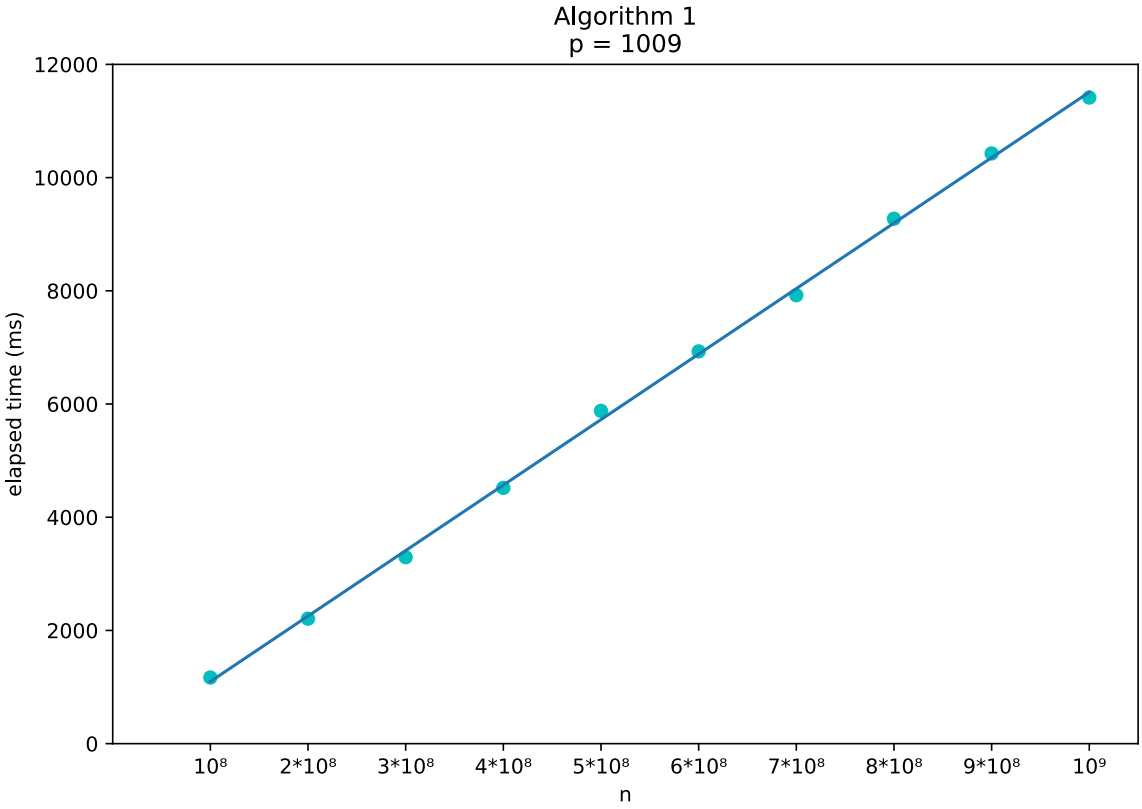
The durations are represented in milliseconds.

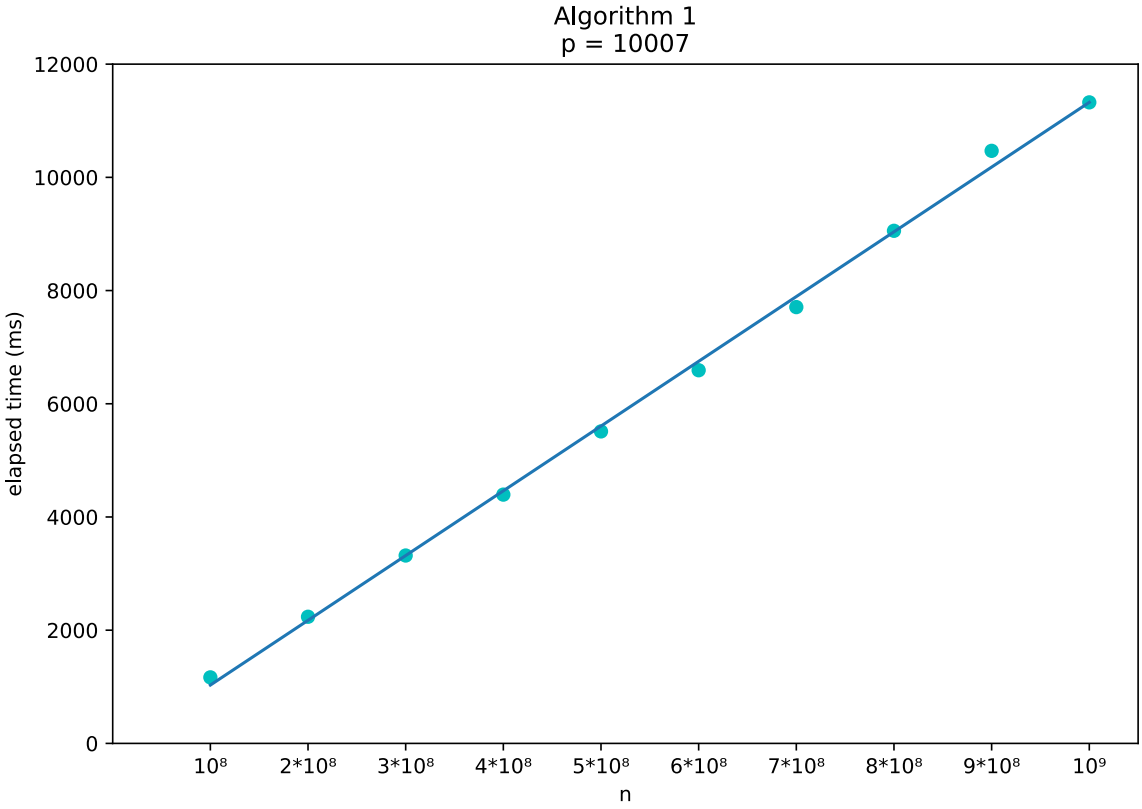
n	Algorithm 1			Algorithm 2			Algorithm 3		
	p=101	p=1009	p=10007	p=101	p=1009	p=10007	p=101	p=1009	p=10007
10^8	1177.35	1168.71	1166.69	0.00165	0.00533	0.06982	0.00097	0.00098	0.00098
$2 * 10^8$	2197.58	2208.14	2237.13	0.00161	0.00523	0.05894	0.00101	0.00101	0.00101
$3 * 10^8$	3523.62	3293.56	3319.27	0.00161	0.00547	0.05987	0.00104	0.00104	0.00104
$4 * 10^8$	4661.67	4518.35	4395.07	0.00160	0.00559	0.05823	0.00105	0.00105	0.00105
$5 * 10^8$	5677.75	5878.80	5511.68	0.00160	0.00566	0.06096	0.00105	0.00105	0.00105
$6 * 10^8$	6744.44	6929.97	6592.31	0.00161	0.00537	0.06044	0.00108	0.00108	0.00108
$7 * 10^8$	7918.60	7921.89	7707.67	0.00160	0.00538	0.05982	0.00108	0.00108	0.00108
$8 * 10^8$	9205.77	9273.15	9055.71	0.00161	0.00559	0.06084	0.00108	0.00108	0.00108
$9 * 10^8$	10142.7	10426.1	10468.3	0.00160	0.00581	0.06259	0.00110	0.00109	0.00109
10^9	11321.9	11413.9	11324.7	0.00160	0.00602	0.06511	0.00108	0.00109	0.00108

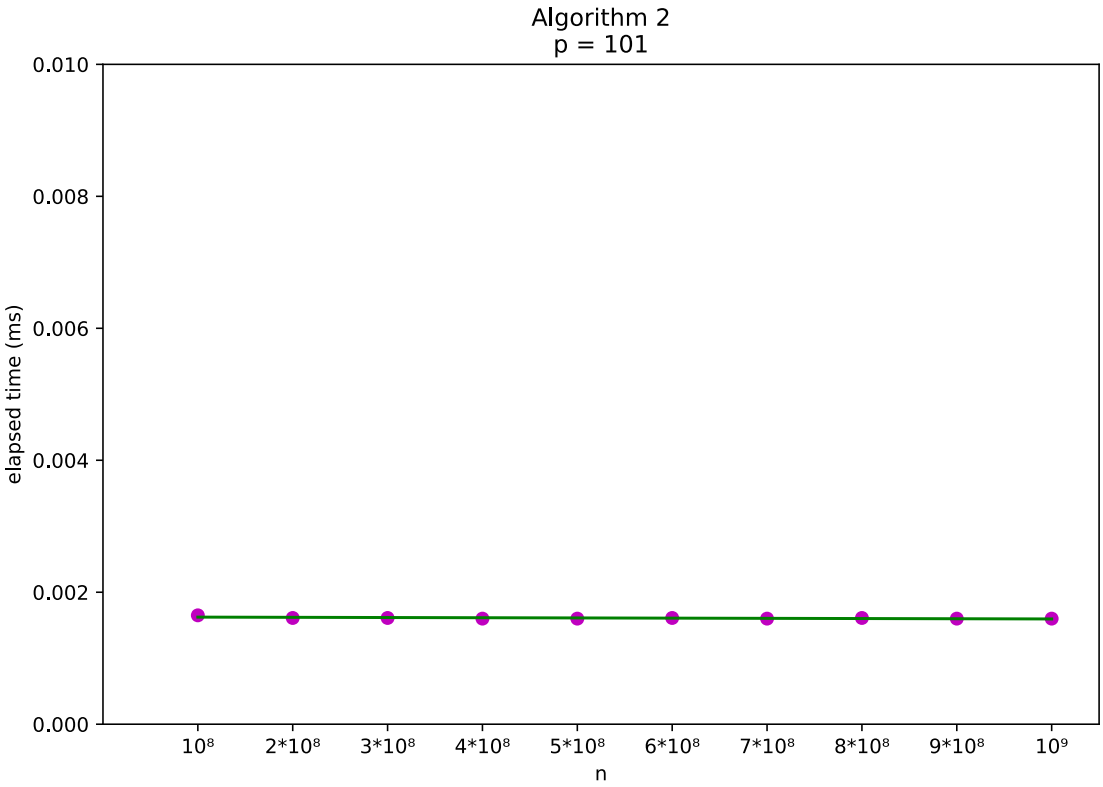
Note: “a” input is assigned to the exact same value for each case, which is 255300.

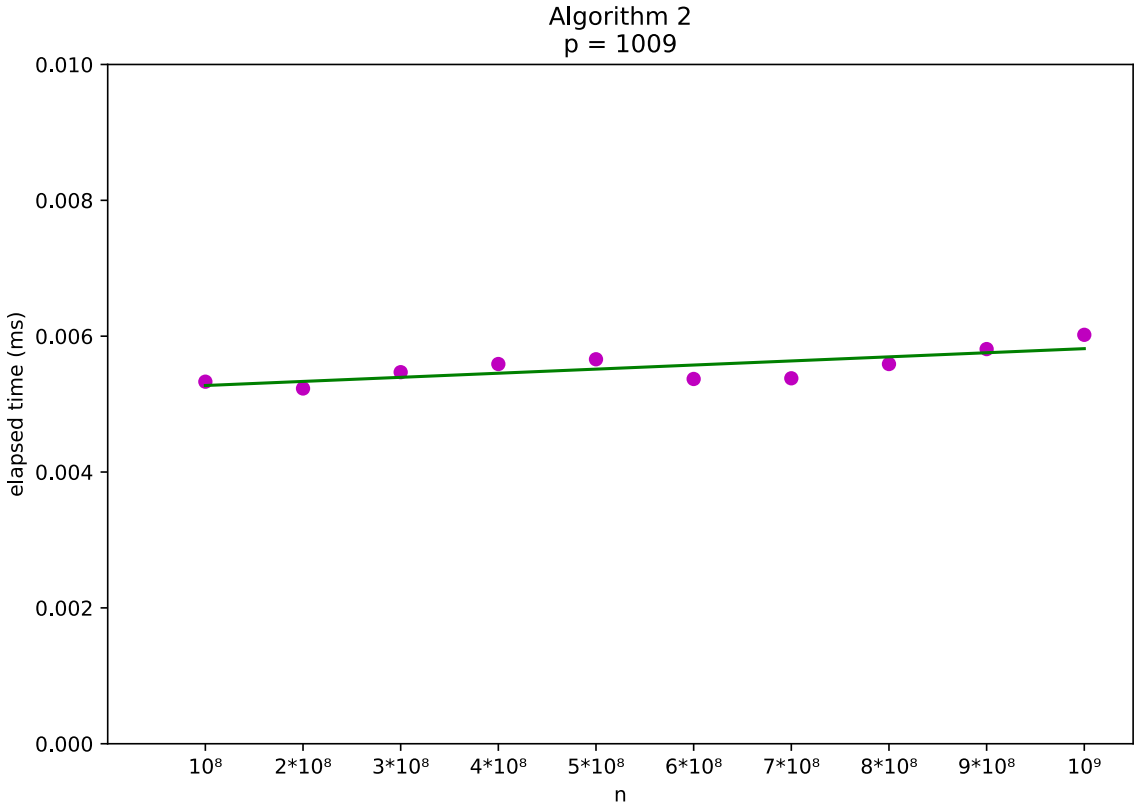
Graphs (Drawn by Python)

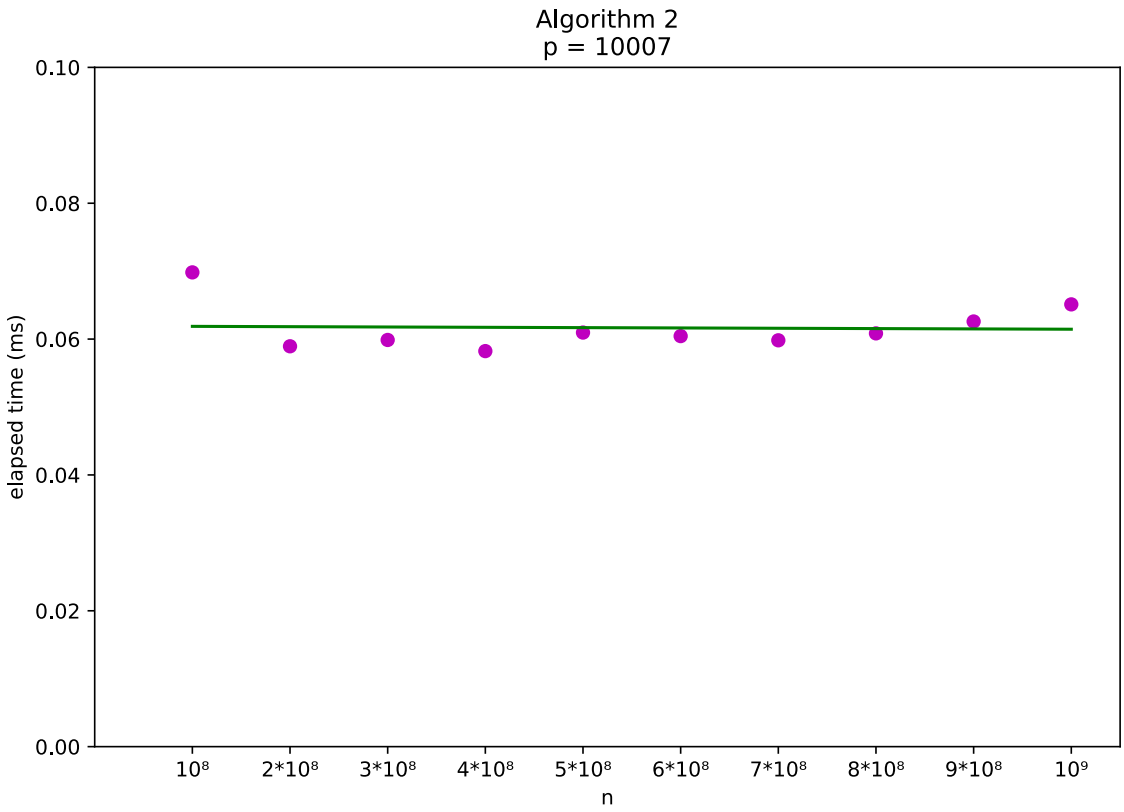


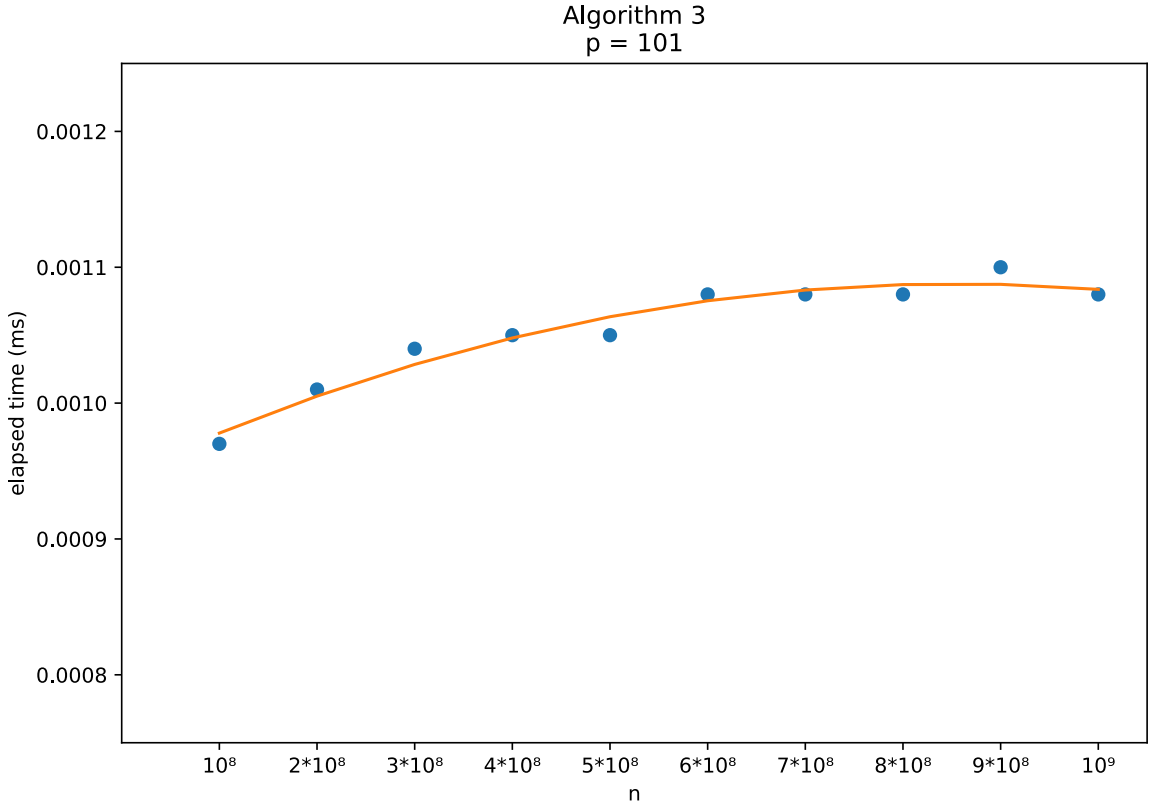

Algorithm 1
p = 1009
elapsed time (ms)
0
2000
4000
6000
8000
10000
12000
10⁸
2*10⁸
3*10⁸
4*10⁸
5*10⁸
6*10⁸
7*10⁸
8*10⁸
9*10⁸
10⁹
n

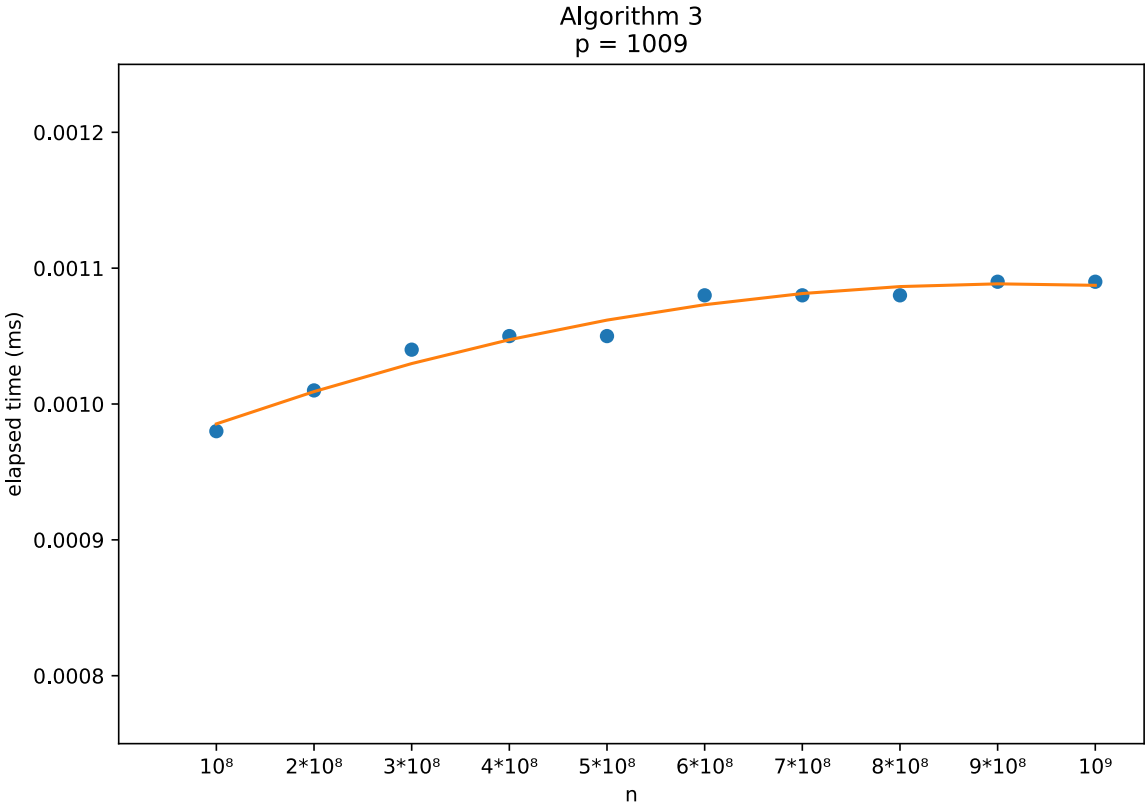

Algorithm 1
p = 10007
elapsed time (ms)
0
2000
4000
6000
8000
10000
12000
10⁸
2*10⁸
3*10⁸
4*10⁸
5*10⁸
6*10⁸
7*10⁸
8*10⁸
9*10⁸
10⁹
n

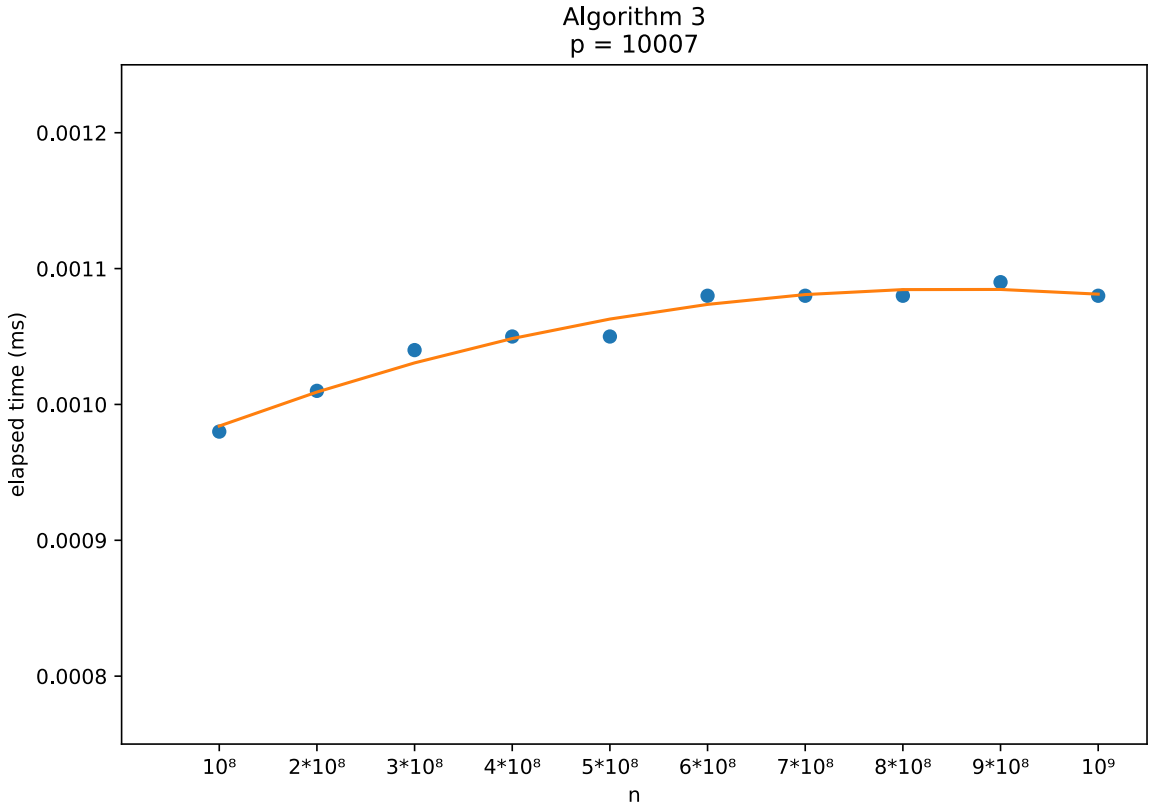

Algorithm 2
p = 101
0.010
0.008
0.006
0.004
0.002
0.000
elapsed time (ms)
10⁸
2*10⁸
3*10⁸
4*10⁸
5*10⁸
6*10⁸
7*10⁸
8*10⁸
9*10⁸
10⁹
n


Algorithm 2
p = 1009
0.010
0.008
0.006
0.004
0.002
0.000
elapsed time (ms)
10⁸
2*10⁸
3*10⁸
4*10⁸
5*10⁸
6*10⁸
7*10⁸
8*10⁸
9*10⁸
10⁹
n


Algorithm 2
p = 10007
elapsed time (ms)
0.10
0.08
0.06
0.04
0.02
0.00
10⁸
2*10⁸
3*10⁸
4*10⁸
5*10⁸
6*10⁸
7*10⁸
8*10⁸
9*10⁸
10⁹
n


Algorithm 3
p = 101
elapsed time (ms)
0.0012
0.0011
0.0010
0.0009
0.0008
10⁸
2*10⁸
3*10⁸
4*10⁸
5*10⁸
6*10⁸
7*10⁸
8*10⁸
9*10⁸
10⁹
n


Algorithm 3
p = 1009
elapsed time (ms)
0.0012
0.0011
0.0010
0.0009
0.0008
10⁸
2*10⁸
3*10⁸
4*10⁸
5*10⁸
6*10⁸
7*10⁸
8*10⁸
9*10⁸
10⁹
n


Algorithm 3
p = 10007
elapsed time (ms)
0.0012
0.0011
0.0010
0.0009
0.0008
10⁸
2*10⁸
3*10⁸
4*10⁸
5*10⁸
6*10⁸
7*10⁸
8*10⁸
9*10⁸
10⁹
n