CS201 Homework-2 Report

Oğuz Akın

22002893

Section-1

Results of all the test runs are represented in the following chart:

$A = 178912$ All results are in milliseconds.	Algorithm 1			Algorithm 2			Algorithm 3		
	$P = 101$	$P = 1009$	$P = 10007$	$P = 101$	$P = 1009$	$P = 10007$	$P = 101$	$P = 1009$	$P = 10007$
$N = 45 * 10^8$	50869	50300	50982	82842	81728	83872	69452	70746	70672
$N = 46 * 10^8$	51586	51348	51821	86063	84791	83935	69813	72892	70729
$N = 47 * 10^8$	52818	52495	52947	87104	87217	86251	71705	72458	72336
$N = 48 * 10^8$	54281	53878	53644	88929	89085	88748	72611	71972	72444
$N = 49 * 10^8$	54937	55587	55182	91822	91166	90557	72282	73342	71516
$N = 50 * 10^8$	56685	56513	56481	91866	91840	93188	72926	73723	72709
$N = 51 * 10^8$	57873	57449	57464	93940	94237	94564	70016	71058	72266
$N = 52 * 10^8$	57939	57860	58372	94525	94674	95833	70771	71002	71349
$N = 53 * 10^8$	59251	59582	59541	97760	96843	97470	70543	70857	73188
$N = 54 * 10^8$	60753	61752	60476	99126	100193	98633	73201	72956	71324

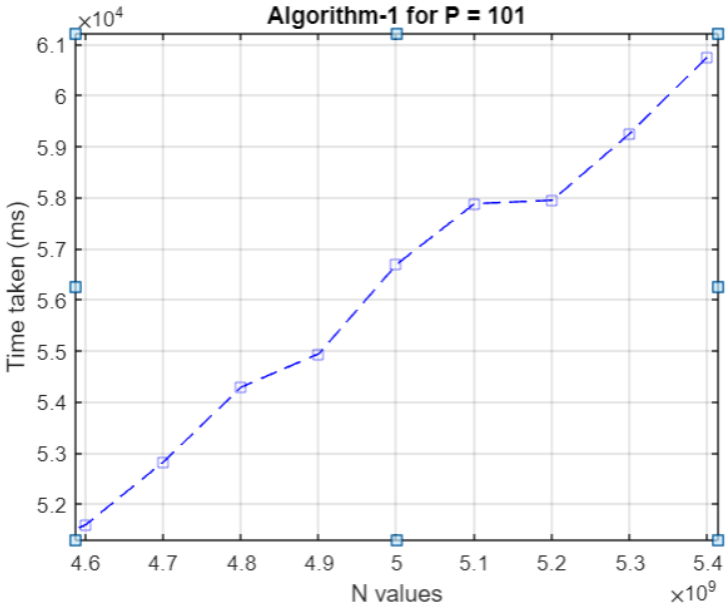
These results were obtained with the computer that has the following specifications:

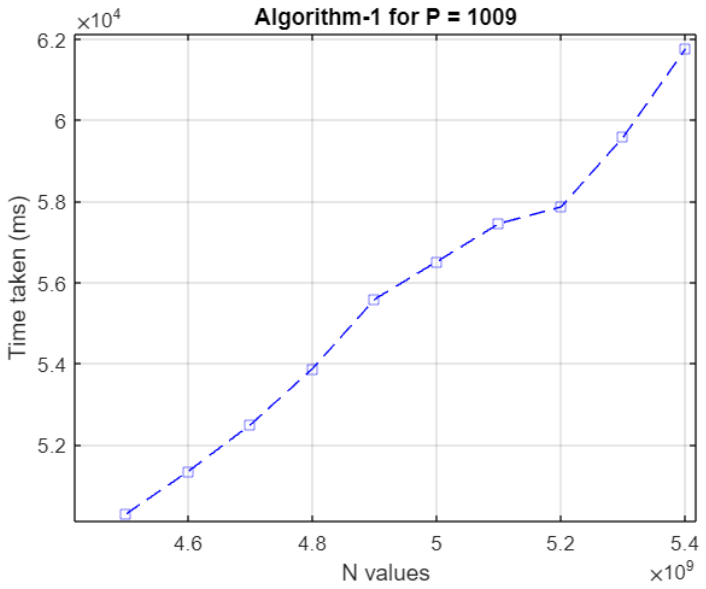
RAM: 16 GB. (Available: 15.9 GB)

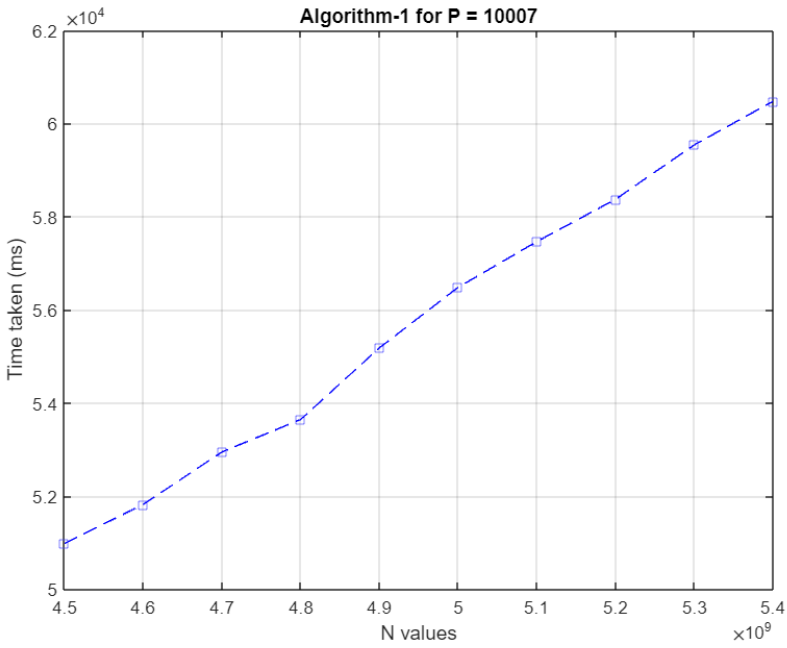
Processor: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz.

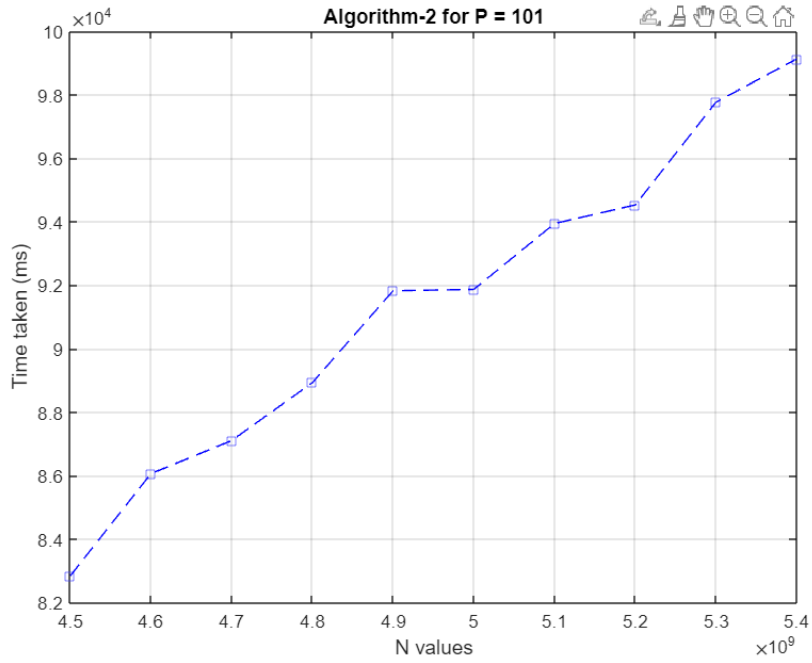
System Type: 64 bit operating system, x64 based processor.

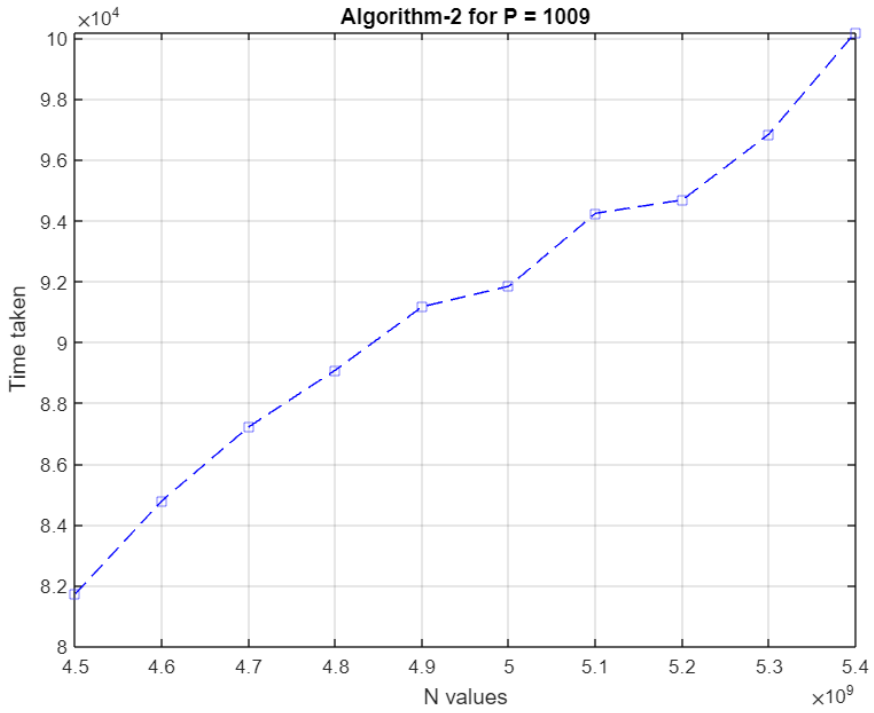
GRAPHS (Time taken is always measured in milliseconds)

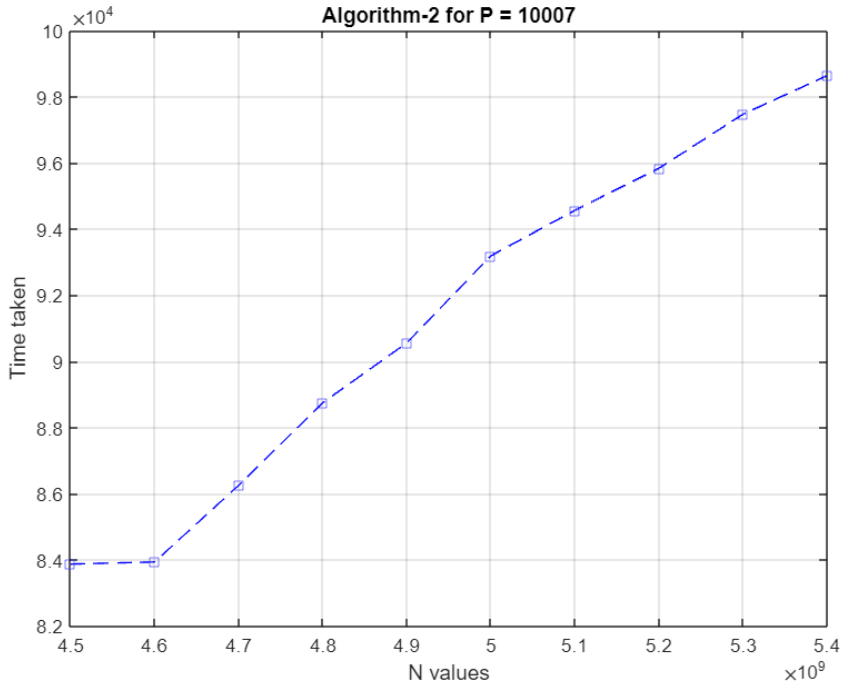


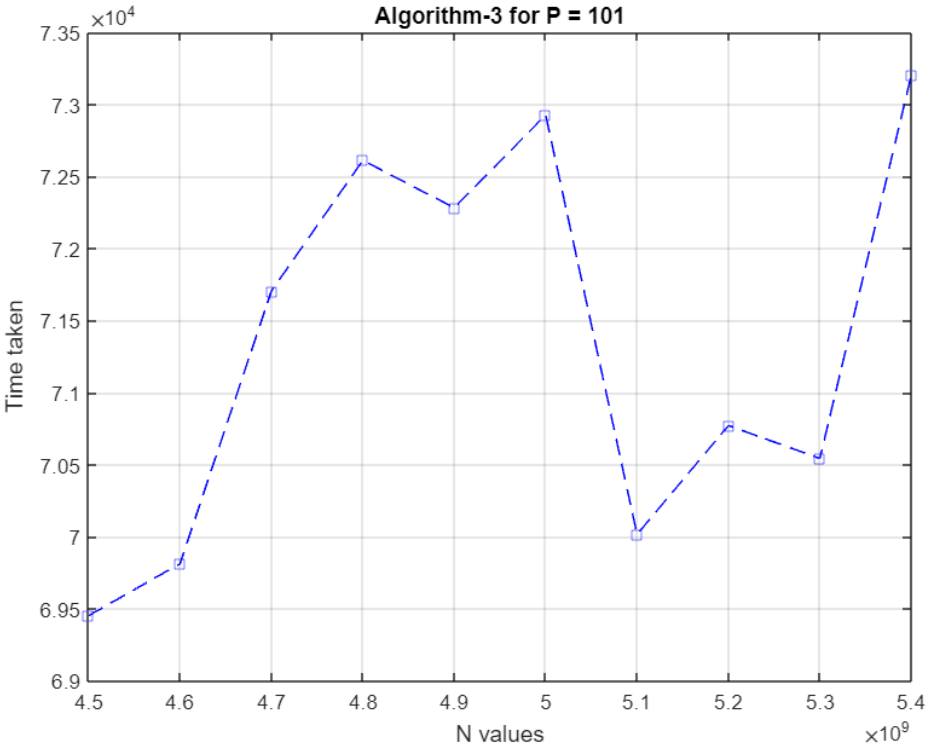


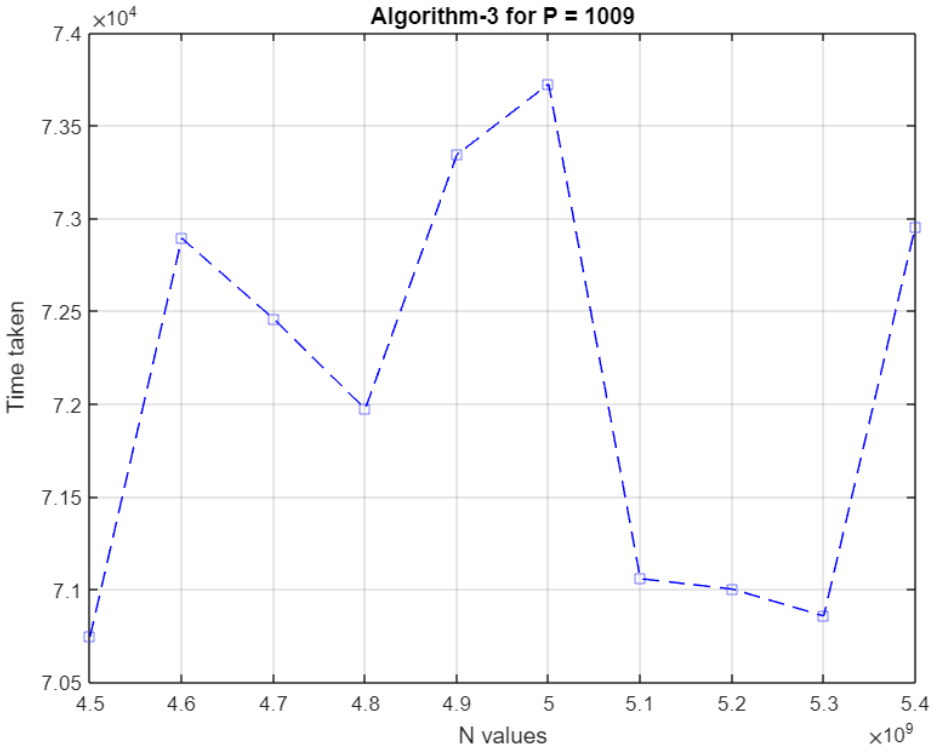

Algorithm-1 for P = 10007
×10^4
6.2
6
5.8
5.6
5.4
5.2
5
Time taken (ms)
4.5
4.6
4.7
4.8
4.9
5
5.1
5.2
5.3
5.4
N values
×10^9

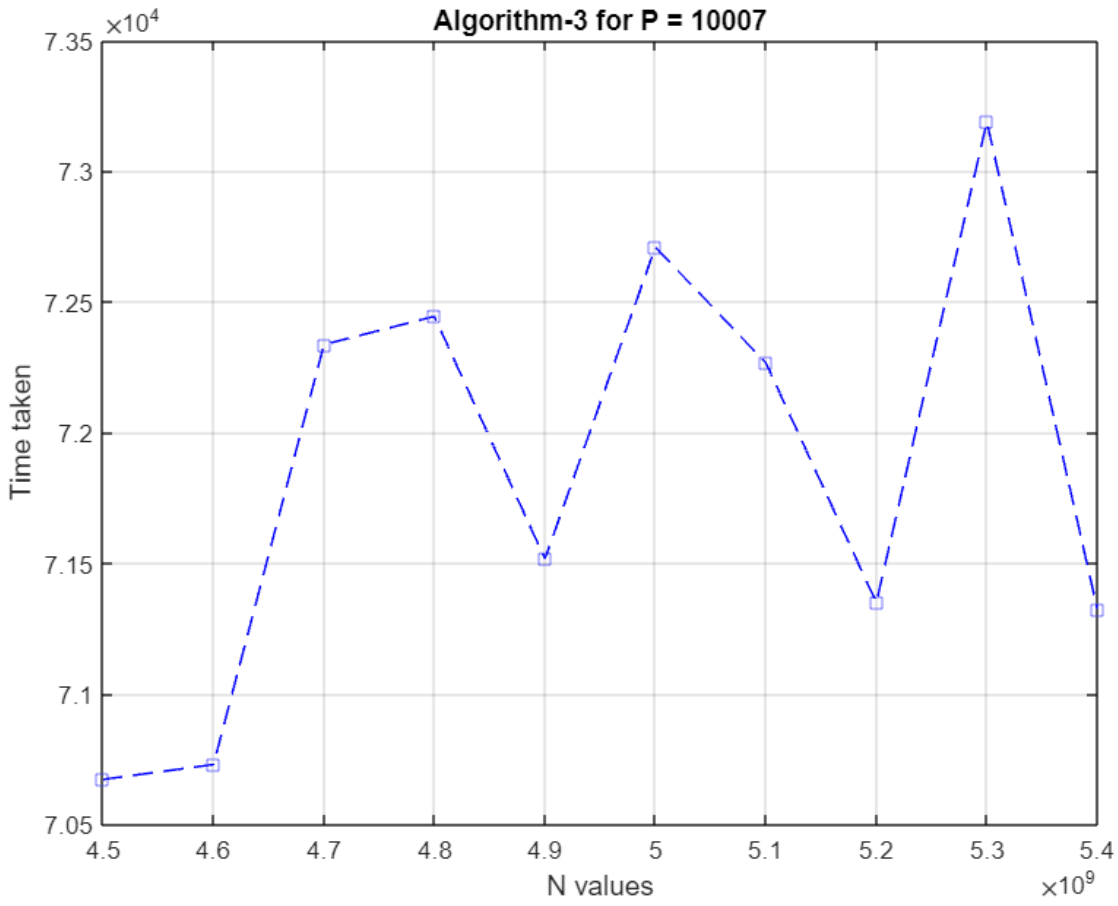

Algorithm-2 for P = 101
×10^4
10
9.8
9.6
9.4
9.2
9
8.8
8.6
8.4
8.2
Time taken (ms)
4.5
4.6
4.7
4.8
4.9
5
5.1
5.2
5.3
5.4
N values
×10^9


Algorithm-2 for P = 1009
×10^4
10
9.8
9.6
9.4
9.2
9
8.8
8.6
8.4
8.2
8
Time taken
4.5
4.6
4.7
4.8
4.9
5
5.1
5.2
5.3
5.4
N values
×10^9


Algorithm-2 for P = 10007
×10^4
10
9.8
9.6
9.4
9.2
9
8.8
8.6
8.4
8.2
Time taken
4.5
4.6
4.7
4.8
4.9
5
5.1
5.2
5.3
5.4
N values
×10^9


Algorithm-3 for P = 101
×10^4
7.35
7.3
7.25
7.2
7.15
7.1
7.05
7
6.95
6.9
Time taken
4.5
4.6
4.7
4.8
4.9
5
5.1
5.2
5.3
5.4
N values
×10^9


Algorithm-3 for P = 1009
×10^4
7.4
7.35
7.3
7.25
7.2
7.15
7.1
7.05
Time taken
4.5
4.6
4.7
4.8
4.9
5
5.1
5.2
5.3
5.4
N values
×10^9



DISCUSSION

Time complexity for the first algorithm is simply calculated by considering the for loop only as the other statements take up O(1) time while the loop takes O(N) time. Time complexity of the second algorithm is similar to the first one but with a slight difference: if there were to be an equation for $a^i \equiv 1 \ (mod\ p)$, it would stop looping and give us the result and thus take less time. However, since we are only considering worst case time complexities, we will imagine that the loop iterates all the way to the end. So, the second algorithm will have time complexity O(N). Time complexity for the third algorithm is different than both its predecessors as this algorithm has a recursive call. Even if the n value were odd or even, the function is called twice to make up the original number and the problem size is always divided into two when passed as a parameter. So, this function has the time complexity $O(2^{\log_2 N})$ which is equal to $O(N)$.