Utku Boran Torun

ID: 21901898

CS201-1 HW2

1. Time Complexity For Each Algorithm

Algorithm 1 (multiplyItself)

For the first algorithm we multiply the int a with itself in a for loop. In each iteration a multiplies itself with a, and take modulus p. This loop executes n times. In the worst case scenario this algorithm's time complexity is O((n-1)p), but since we do not include constants in our time complexity represantation we say that O(np) is our worst case time complexity. In best case scenario time complexity is O(1) and the average case is O((n/2) – 1)p), but we still say that average case time complexity is O(np).

Algorithm 2 (cycleShortcut)

Second algorithm is bit different than the first one. First it finds the i which satisfies a^i=1(mod p), then calculates a^(n mod(i)) mod(p) which will give us at the conclusion a^n mod(p). If n < i our time complexity will be same with the previous algorithm (multiplyItself) which is O(np), else our time complexity will be different. If a % p equals 1 at the beginning then time complexity is same as O(np), but other than that its time complexity is O(p + n).

Algorithm 3 (recursive)

This algorithm is recursive algorithm. We compute different results based on whether n is even or not. Time complexity is O(plogn) in worst case scenario. Best case is O(1) and still average case is O(plogn).

2. Specifications Of PC (My computer has problem with its battery)

Processor: 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz 2.30 GHz

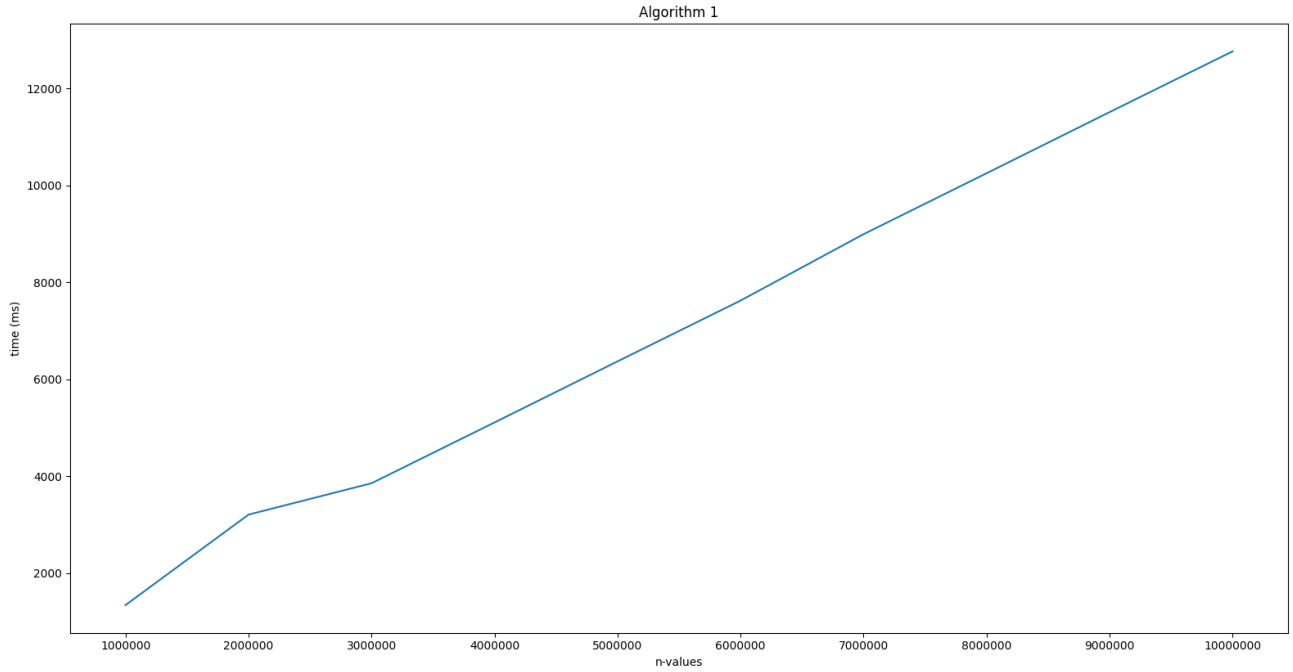
Ram: 16,0 GB

3. Runtime Chart (Table)

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
10^6	1345 ms	12235 ms	110476 ms	1.578 ms	10.384 ms	14.965 ms	0.2012 ms	2.0783 ms	19.147 ms
$2*10^6$	3211 ms	24754 ms	210894 ms	3.783 ms	11.872 ms	15.842 ms	0.2093 ms	2.1007 ms	19.537 ms
$3*10^6$	3858 ms	37276 ms	311412 ms	4.984 ms	13.365 ms	16.711 ms	0.2177 ms	2.1222 ms	19.936 ms
$4*10^6$	5112 ms	49783 ms	411920 ms	5.190 ms	14.849 ms	17.600 ms	0.2256 ms	2.1455 ms	20.342 ms
$5*10^6$	6372 ms	62311 ms	512428 ms	6.397 ms	16.331 ms	18.476 ms	0.2356 ms	2.1680 ms	20.757 ms
$6*10^6$	7626 ms	74834 ms	612936 ms	7.502 ms	17.829 ms	19.355 ms	0.2451 ms	2.1907 ms	21.181 ms
$7*10^6$	8995 ms	87352 ms	713444 ms	8.807 ms	19.312 ms	20.227 ms	0.2550 ms	2.2137 ms	21.613 ms
$8*10^6$	10252 ms	99873 ms	813950 ms	10.012 ms	20.800 ms	21.104 ms	0.2653 ms	2.2369 ms	22.054 ms
$9*10^6$	11513 ms	112392 ms	914458 ms	11.217 ms	22.288 ms	21.981 ms	0.2759 ms	2.2604 ms	22.504 ms
$10*10^6$	12766 ms	124911 ms	1014966 ms	12.422 ms	23.776 ms	22.858 ms	0.2871 ms	2.2841 ms	22.963 ms

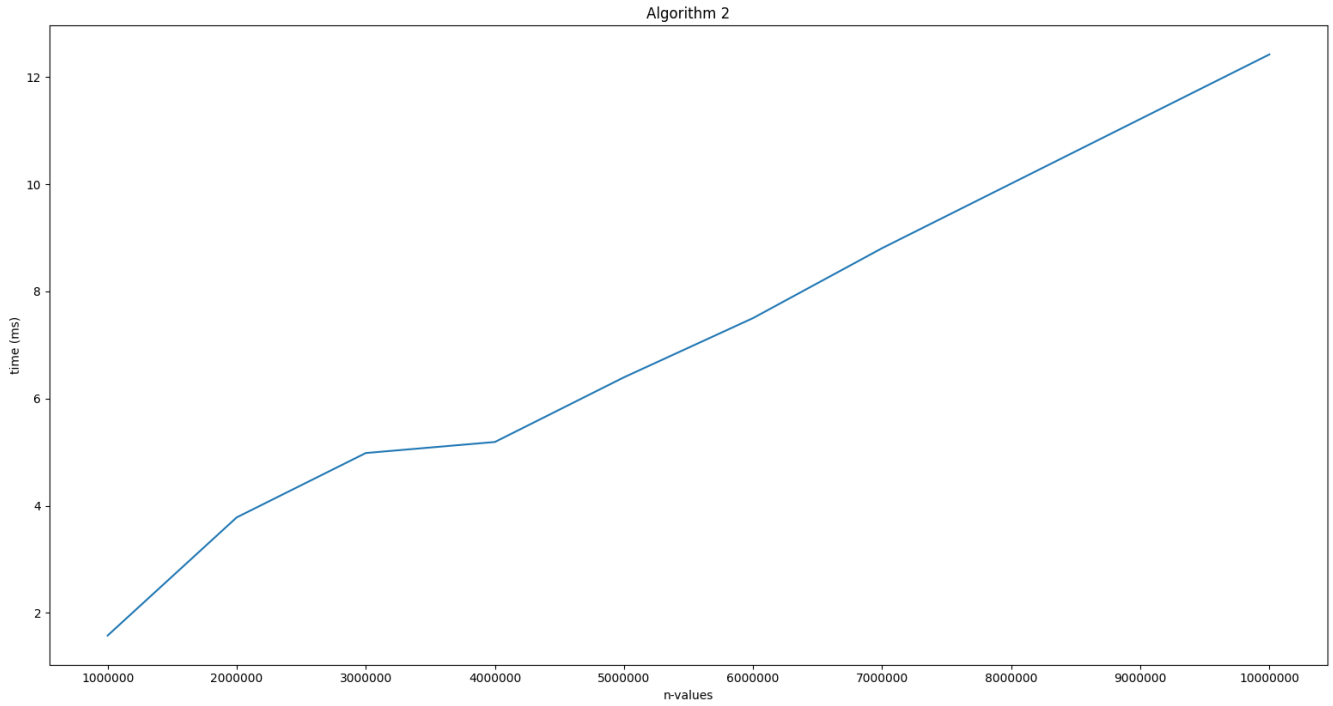
4. Plotting Of Algorithms

Algorithm 1



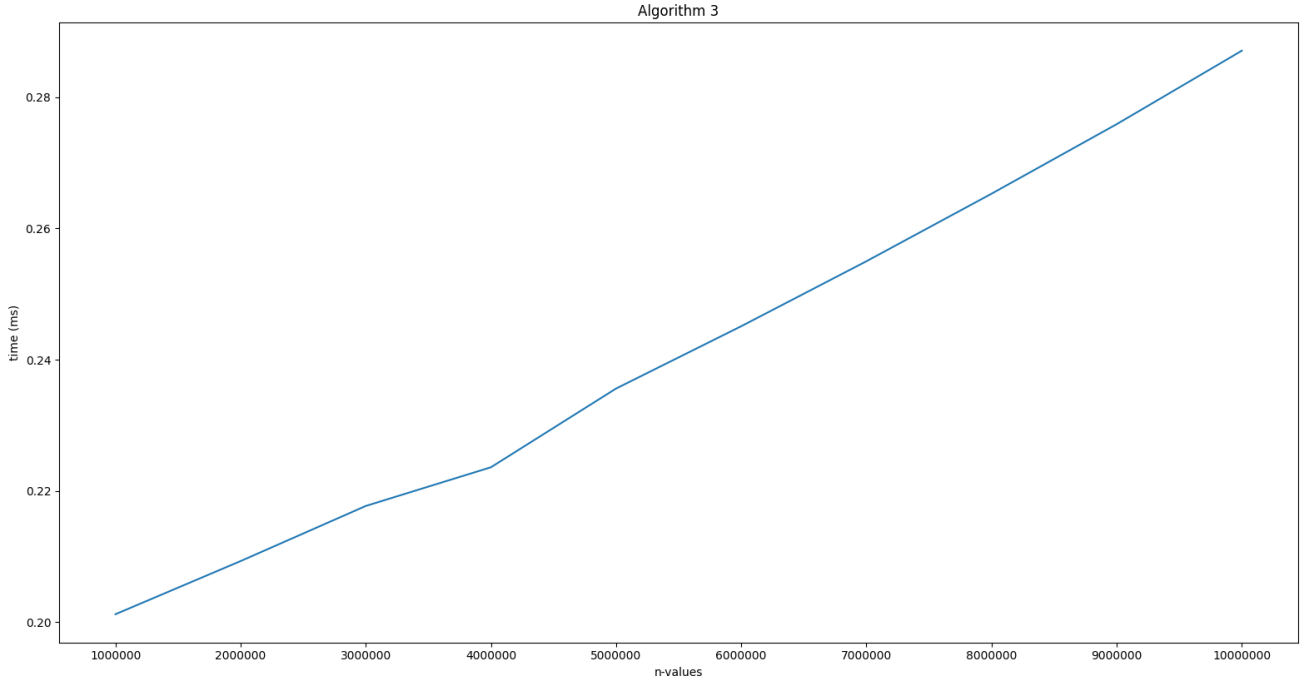
As we can see from the graph as well this algorithm's time complexity is linear.

Algorithm 2



This algorithm’s time complexity is linear as well, but there is fluctuations at the beginning of the smaller n-values

Algorithm 3



This algorithm’s time complexity is not linear, but it is logarithmic. In addition there is little fluctuations at the beginning because of the small n-values.