

Assignment 1

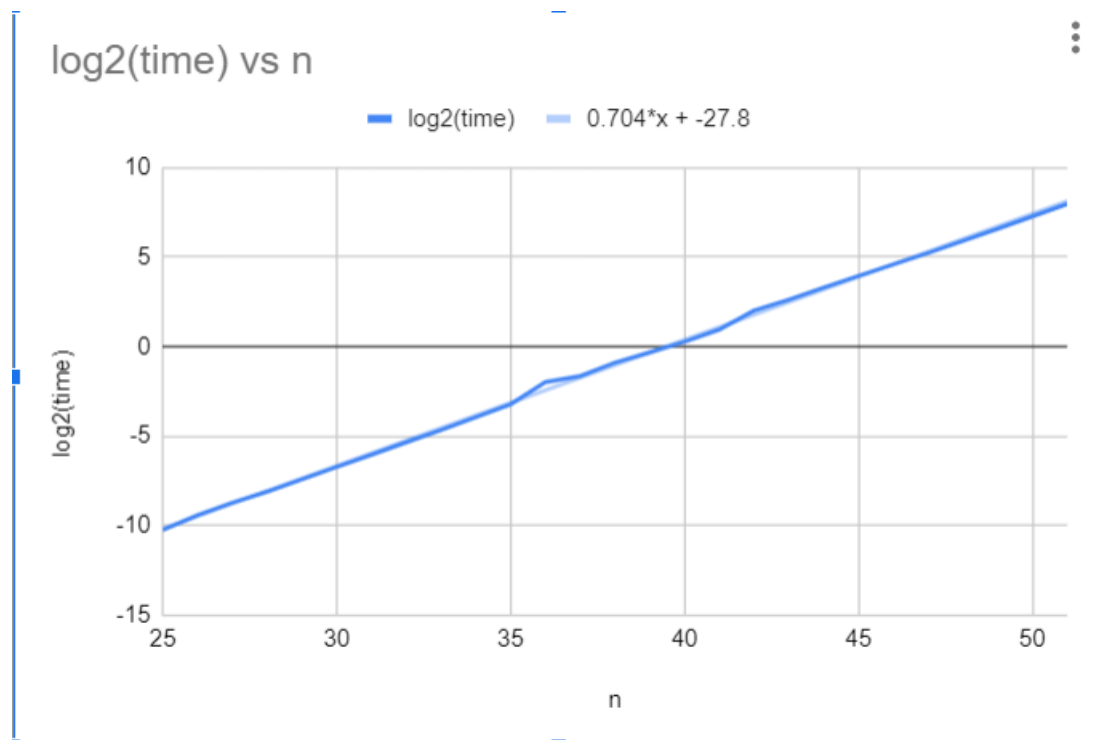
Q1.

Time(s)	0.001	0.1	1	5	60	600
Rfib(n)	26	34	40	43	48	53
lfib(n)	$2.1 \cdot 10^5$	$2.1 \cdot 10^7$	$2.08 \cdot 10^8$	$1.1 \cdot 10^9$	$2 \cdot 10^{10}$	$2 \cdot 10^{11}$
Cleverfib(n)	$>10^{18}$	$>10^{18}$	$>10^8$	$>10^{18}$	$>10^{18}$	$>10^{18}$

Q2.

i)Rfib(n)

n	time(S)	log2(time)
25	0.000834	-10.227665
26	0.001471	-9.408987038
27	0.002396	-8.705156377
28	0.003645	-8.09986547
29	0.005871	-7.412178028
30	0.009477	-6.721353847
31	0.015284	-6.031834027
32	0.02471	-5.338761179
33	0.039781	-4.651776647
34	0.065021	-3.942950445
35	0.107342	-3.219713421
36	0.254536	-1.974058378
37	0.321351	-1.637778133
38	0.540364	-0.887996532
39	0.801826	-0.318638896
40	1.207907	0.272509382
41	1.93719	0.953965461
42	4.04712	2.016895627
43	6.193032	2.630645901
44	9.835174	3.297950576
45	15.48638	3.95292795
47	37.53292	5.230084744
49	95.71717	6.580705898
51	2.510058	7.971576736

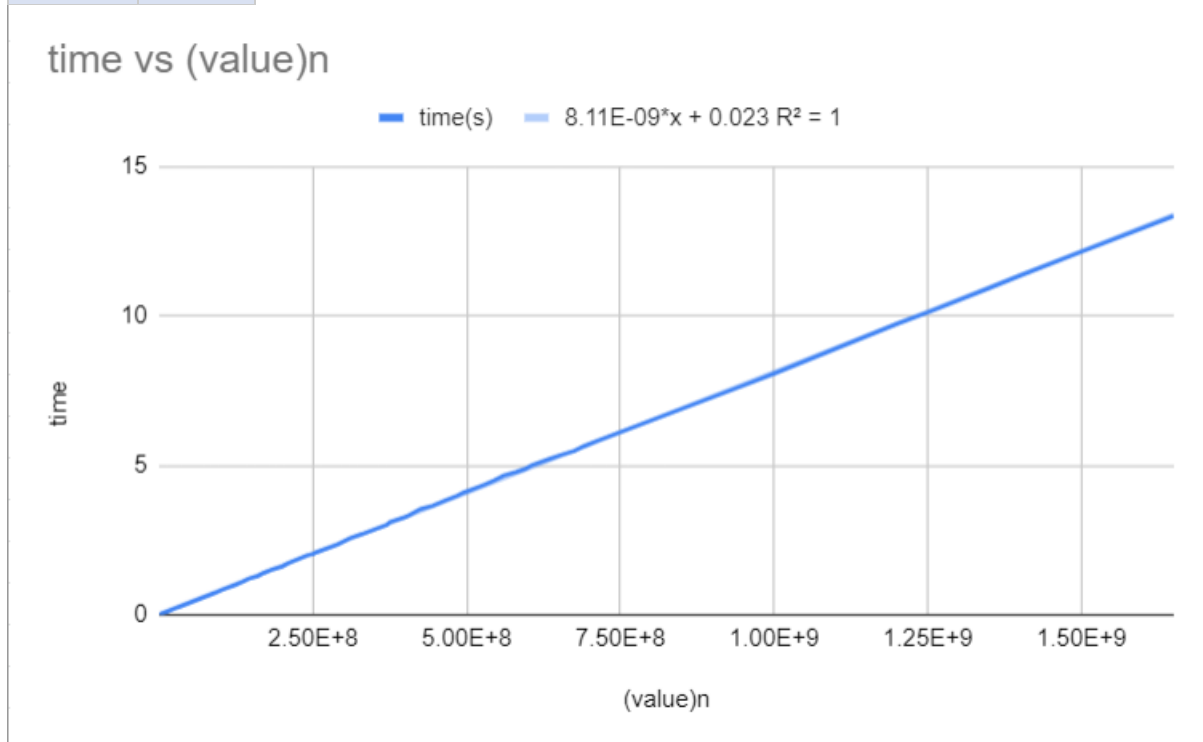


ii)(IFIB)

(value)n	time
200000	0.001703
520000	0.043459
1020000	0.08398
1520000	0.124013
2520000	0.209369
3520000	0.287753
4520000	0.366598
5520000	0.447184
6020000	0.486996
6520000	0.527032
7020000	0.568071
7520000	0.607086
9020000	0.727729
10020000	0.814987
11020000	0.894929
12020000	0.971608
12520000	1.011615
13020000	1.058381

135200000	1.100582
145200000	1.204486
160200000	1.295128
170200000	1.400957
185200000	1.524827
200200000	1.61828
210200000	1.725975
240200000	1.987983
245200000	2.013307
250200000	2.028869
255200000	2.087233
290200000	2.346973
310200000	2.568261
330200000	2.702752
365200000	2.967797
370200000	3.004562
375200000	3.09916
400200000	3.271152
405200000	3.319148
425200000	3.54061
445200000	3.652743
465200000	3.830737
485200000	3.983546
495200000	4.091975
520200000	4.288054
530200000	4.365929
545200000	4.50153
560200000	4.661183
580200000	4.775569
595200000	4.882017
605200000	4.999565
610200000	5.037631
635200000	5.220591
655200000	5.360182
675200000	5.477107

690200000	5.636898
710200000	5.808593
850000000	6.889823
950000000	7.672583
1000000000	8.067554
1200000000	9.736658
1250000000	10.117065
1350000000	10.933563
1400000000	11.35074
1500000000	12.149596
1650000000	13.339017

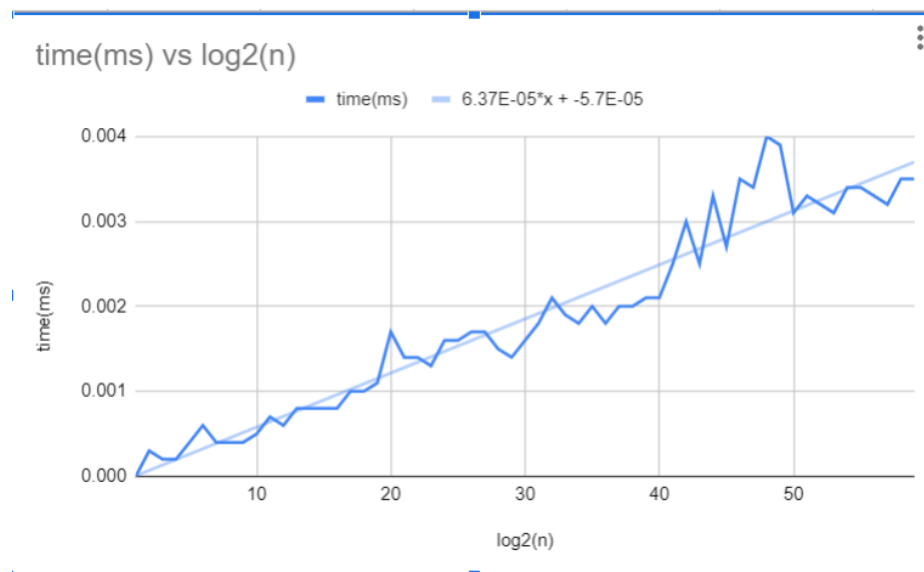


iii) cleverfib(n)

log ₂ (n)	time(ms)
1	0
2	0.0003
3	0.0002
4	0.0002

5	0.0004
6	0.0006
7	0.0004
8	0.0004
9	0.0004
10	0.0005
11	0.0007
12	0.0006
13	0.0008
14	0.0008
15	0.0008
16	0.0008
17	0.001
18	0.001
19	0.0011
20	0.0017
21	0.0014
22	0.0014
23	0.0013
24	0.0016
25	0.0016
26	0.0017
27	0.0017
28	0.0015
29	0.0014
30	0.0016
31	0.0018
32	0.0021
33	0.0019
34	0.0018
35	0.002
36	0.0018

37	0.002
38	0.002
39	0.0021
40	0.0021
41	0.0025
42	0.003
43	0.0025
44	0.0033
45	0.0027
46	0.0035
47	0.0034
48	0.004
49	0.0039
50	0.0031
51	0.0033
52	0.0032
53	0.0031
54	0.0034
55	0.0034
56	0.0033
57	0.0032
58	0.0035
59	0.0035



- a) i) graph of $\log_2(\text{time})$ vs n for Rfib is linear as the total number of instructions executed for the computation as Rfib(n) is $>2^{(n-1)}/2$. the total number of instructions executed for Ifib(n) is $3*n$ and hence the graph of time vs n is linear. The total Number of instructions executed for Cleverfib(n) is proportional to $\log(n-1)$ so the graph of time vs $\log_2(n)$ is linear.
- b) Slope of Rfib: 0.704 ($\log_2(\text{time}(s))$ vs n)
- slope of Ifib: 8.11×10^{-9} (time(s) vs n)
- Slope of Clever fib: 6.37×10^{-5} (time(ms) Vs $\log(n)$)

The difference in slope is due to the difference in the number of instructions in each iteration of a specific function.

- c) 1) yes this will affect the running time of cleverfib. Running time per iteration for clever fib is significantly more as compared to Ifib and Rfib. This we can verify from the slope of graphs plotted for Ifib and Cleverfib (because they are of the same order). Value of slope of cleverfib is more than that of Ifib
- 2) no, these facts have negligible effect on the relative speed of clever fib as the number of total instructions executed for computation when we call clever fib is still far far lesser as compared to the other two algorithm.
- 3) yes, yes (word Ram model of computation gives us a good idea about the efficiency of an algorithm by giving the total number of instructions executed and thus is quite accurate in comparing the efficiency of a pair of algorithms, however, it does not accurately give us the running time of an algorithm because there are various other factors that affect the running time .)