

CS 201 HOMEWORK 2 ALGORITHMS ANALYSIS

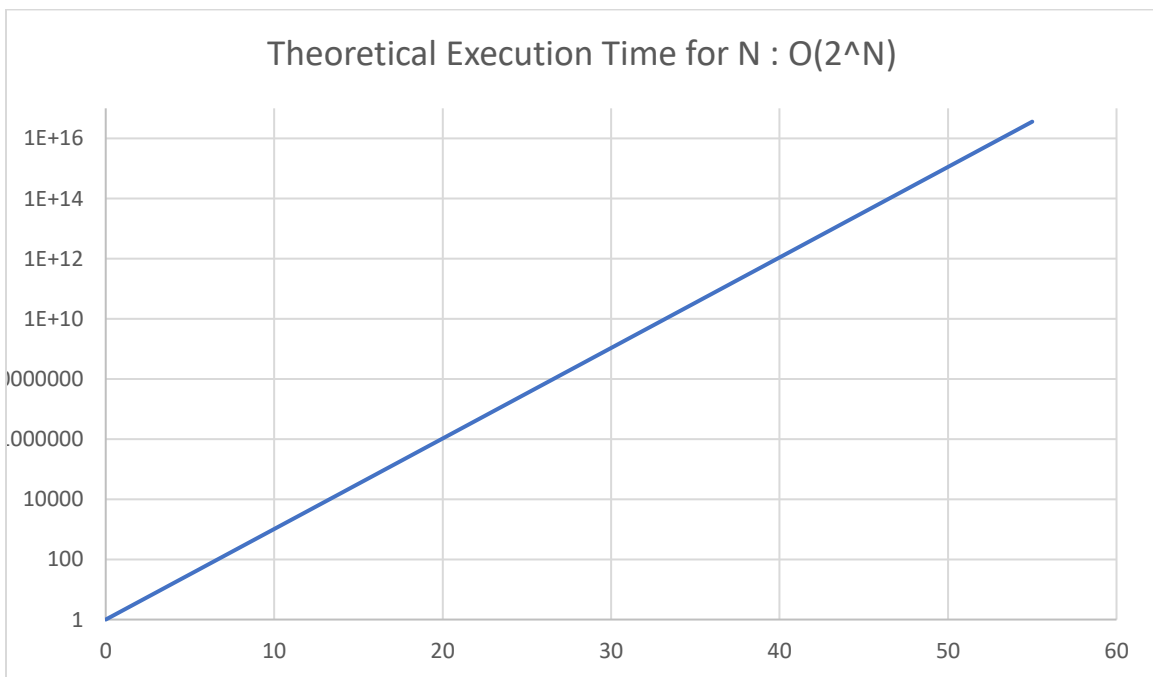
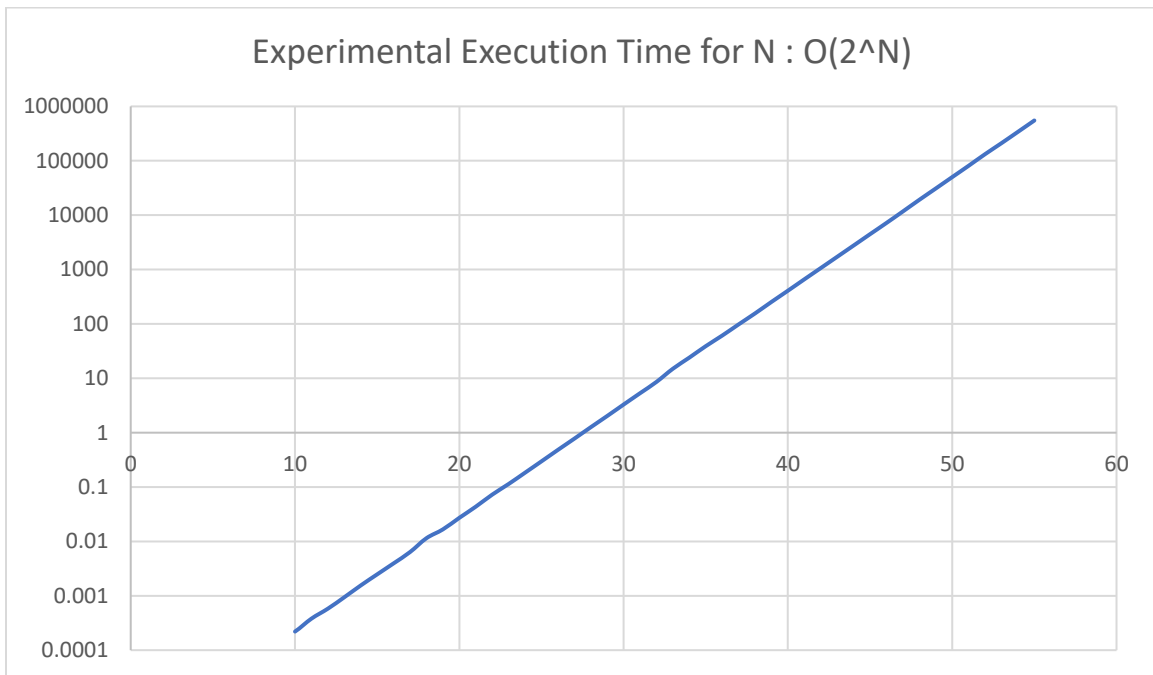
RECURSIVE SOLUTION TO FIBONACCI SEQUENCE

Table for Recursive Solution

N	Execution Time for N (ms)
9	0.00015
10	0.00022
11	0.00038
12	0.00058
13	0.00094
14	0.00155
15	0.00248
16	0.00395
17	0.00637
18	0.01146
19	0.01665
20	0.02715
21	0.04349
22	0.07248
23	0.11376
24	0.18356
25	0.29688
26	0.48211
27	0.77521
28	1.25943
29	2.02593
30	3.2923
31	5.31675
32	8.58732
33	15
34	24
35	39
36	61
37	98
38	156

39	254
40	408
41	661
42	1065
43	1717
44	2768
45	4474
46	7198
47	11710
48	19175
49	30940
50	50145
51	81037
52	132397
53	211114
54	340772
55	550824

Corresponding Graphs for Recursive Solution



ITERATIVE SOLUTION TO FIBONACCI SEQUENCE

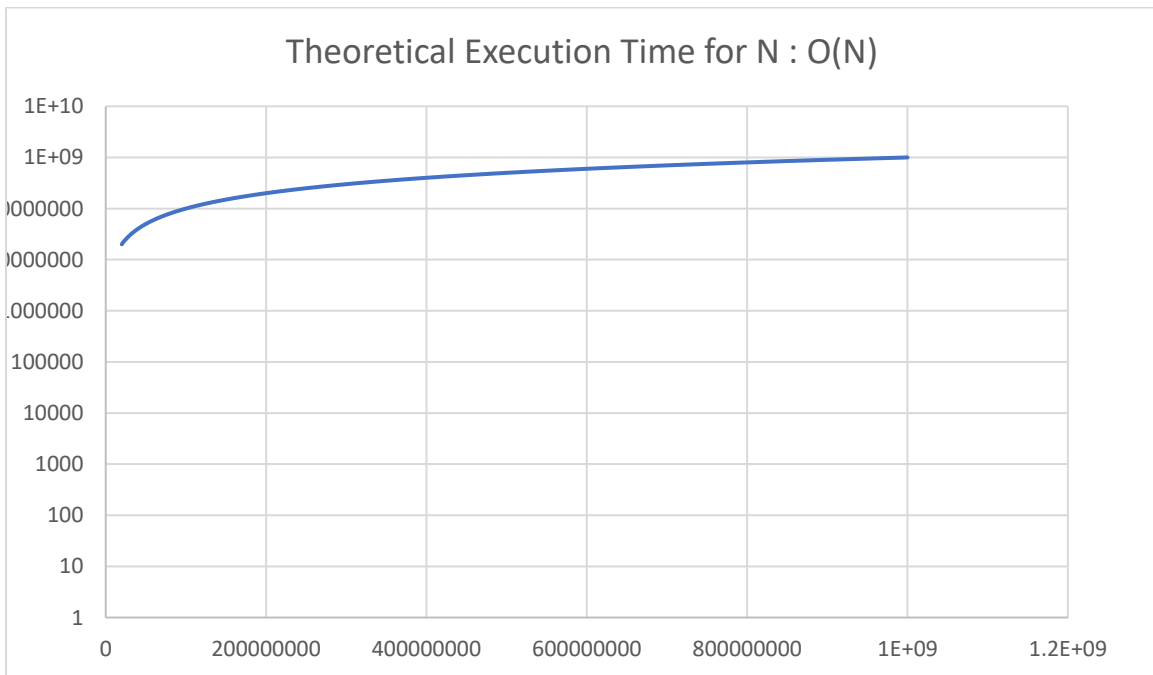
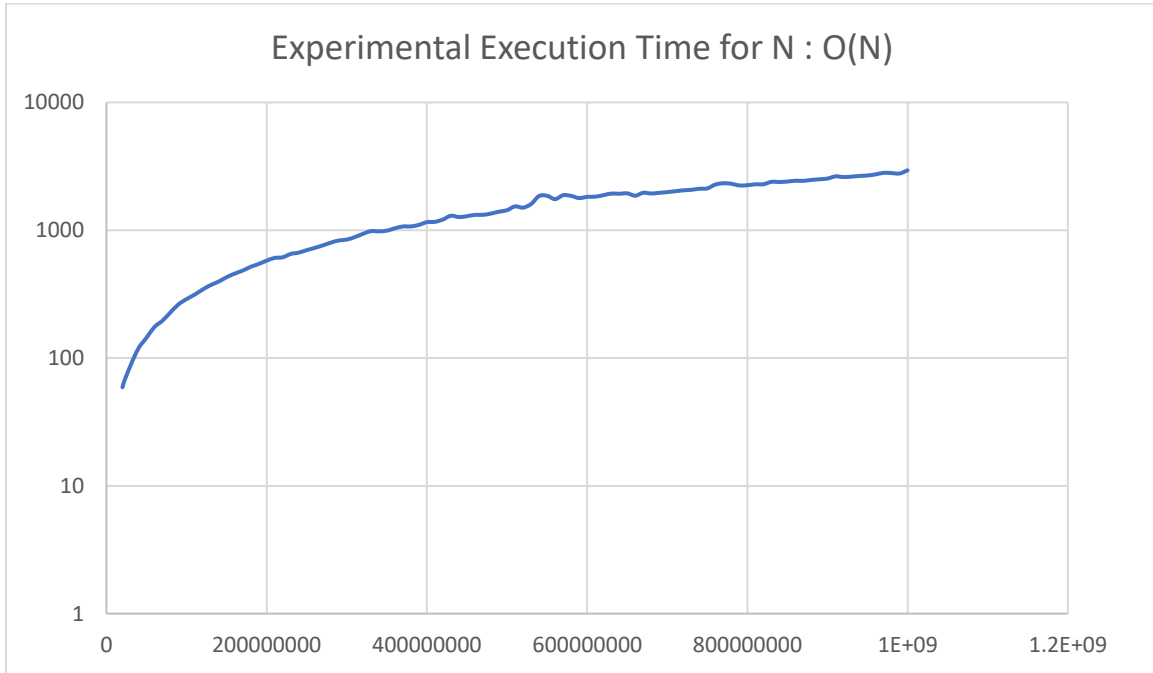
Table for Iterative Solution

N	Execution Time for N (ms)
10000001	31
20000001	59
30000001	87
40000001	119
50000001	144
60000001	175
70000001	196
80000001	228
90000001	263
100000001	289
110000001	313
120000001	343
130000001	372
140000001	396
150000001	429
160000001	457
170000001	483
180000001	517
190000001	544
200000001	578
210000001	607
220000001	614
230000001	652
240000001	668
250000001	698
260000001	727
270000001	760
280000001	799
290000001	831
300000001	845
310000001	883
320000001	936

330000001	983
340000001	980
350000001	992
360000001	1035
370000001	1070
380000001	1071
390000001	1101
400000001	1156
410000001	1163
420000001	1213
430000001	1298
440000001	1267
450000001	1288
460000001	1319
470000001	1319
480000001	1350
490000001	1397
500000001	1434
510000001	1536
520000001	1503
530000001	1602
540000001	1852
550000001	1860
560000001	1751
570000001	1879
580000001	1856
590000001	1784
600000001	1825
610000001	1829
620000001	1879
630000001	1935
640000001	1929
650000001	1944
660000001	1863
670000001	1964
680000001	1938

6900000001	1963
7000000001	1986
7100000001	2019
7200000001	2052
7300000001	2071
7400000001	2111
7500000001	2124
7600000001	2276
7700000001	2330
7800000001	2309
7900000001	2239
8000000001	2251
8100000001	2287
8200000001	2290
8300000001	2391
8400000001	2385
8500000001	2403
8600000001	2439
8700000001	2434
8800000001	2476
8900000001	2504
9000000001	2536
9100000001	2639
9200000001	2606
9300000001	2626
9400000001	2659
9500000001	2681
9600000001	2734
9700000001	2812
9800000001	2799
9900000001	2779
10000000001	2941

Corresponding Graphs for Iterative Solution



Computer Specifications

- Intel core i7 7700-HQ
- Nvidia GTX 1060
- 256gb ssd
- 1TB hdd
- 16gb ram
- 64-bit
- Dell 7577

Comparison

Recursion is a strong tool in solving some problems however, there are some cases such as Fibonacci Sequences in which recursion is not performing very well like normal or iterative solutions. In Fibonacci sequences, on the one hand, the iterative solution has $T(\text{iterativeFib}) = O(\text{iterativeFib}) = O(N)$, on the other, the recursive solution has $T(\text{fib}) = O(\text{fib}) = O(2^N)$. These results indicate that the iterative solution is much more efficient. According to the provided tables, with the iterative solution, huge Fibonacci numbers are calculatable, whereas, with the recursive solution, even the first 55 Fibonacci numbers took a huge amount of time compared the iterative method. The reason of it is that the recursive algorithm grows exponentially. Its increase is much larger than the iterative solution which grows linearly. Recursion should be used only when it makes a problem much easier to solve and when it makes the solution much more efficient. In today's world, having fast algorithms is highly important. It is even more important than storage because there is enough storage in our hardware's. Everybody values fast algorithms.