

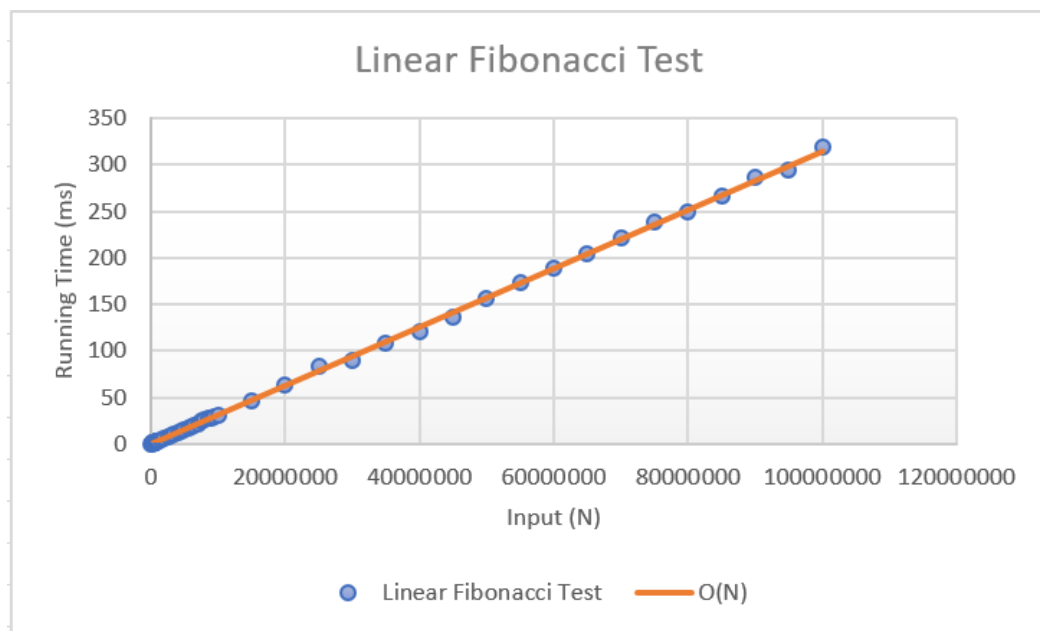
CS 201 – HW2

LINEAR ALGORITHM DATA

Input (N)	Running Time (ms)
100000	0
150000	0
200000	1
250000	1
300000	1
350000	1
400000	1
450000	2
500000	2
550000	1
600000	1
650000	2
700000	2
750000	3
800000	3
850000	3
900000	3
950000	3
1000000	3
1500000	5
2000000	6
2500000	8
3000000	9
3500000	11
4000000	12
4500000	14
5000000	16
5500000	17
6000000	19
6500000	20
7000000	22
7500000	25
8000000	26
8500000	27

9000000	28
9500000	30
10000000	31
15000000	47
20000000	63
25000000	83
30000000	90
35000000	109
40000000	121
45000000	137
50000000	157
55000000	174
60000000	189
65000000	205
70000000	222
75000000	239
80000000	250
85000000	267
90000000	286
95000000	295
100000000	319

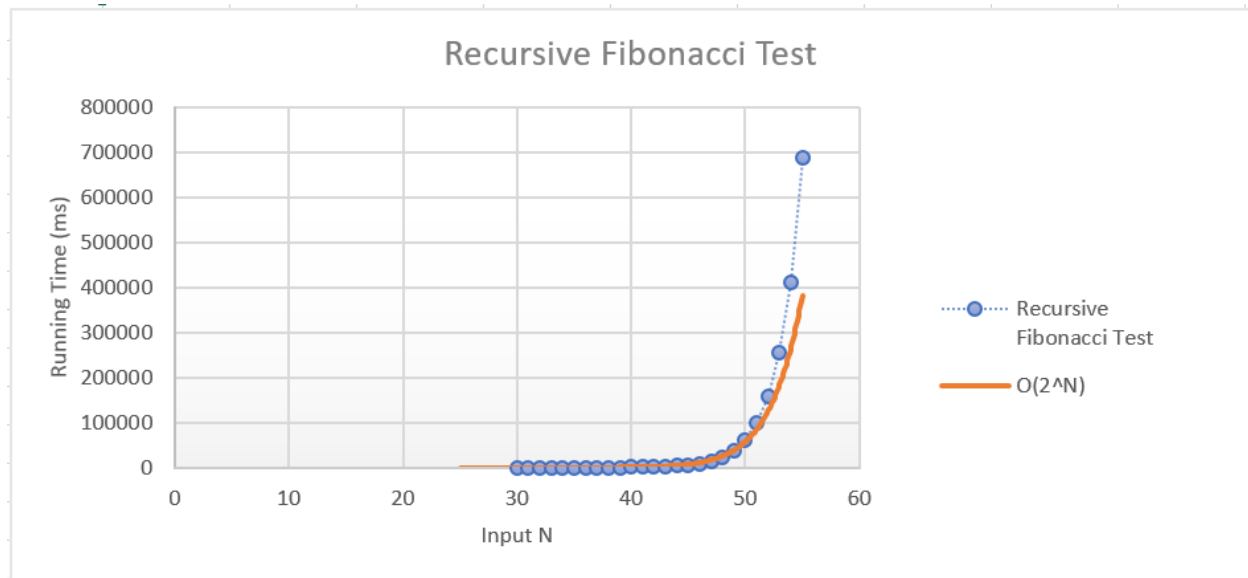
LINEAR ALGORITHM GRAPH



RECURSIVE ALGORITHM DATA

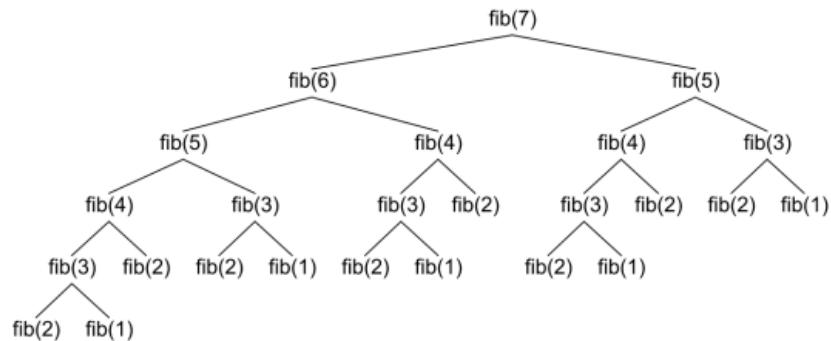
Input (N)	Running Time (ms)
30	4
31	7
32	11
33	16
34	28
35	45
36	72
37	116
38	193
39	305
40	501
41	800
42	1301
43	2088
44	3367
45	5436
46	8797
47	14260
48	23030
49	37366
50	60662
51	97738
52	157410
53	255133
54	412568
55	688636

RECURSIVE ALGORITHM GRAPH



DISCUSSION

Linear Fibonacci calculation algorithm was expected to have $O(N)$ time complexity because it includes a single size N loop. Recursive Fibonacci calculation algorithm was expected to have $O(2^N)$ time complexity because it creates more # of function calls than needed, as below:



This was the reason why recursive algorithm may take more than days/weeks or the time between now and the Big Bang.

Overall, the results of the assignment match closely with these theoretical assumptions.

Selim Firat Yilmaz
21502736

COMPUTER SPECIFICATIONS

OS: Windows 10 64-Bit

Processor: Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz, 2601 Mhz, 2 Core(s), 4 Logical Processor(s)

RAM: 8 GB

Compiled in g++ and executed in command line.