Judging from the results of the experiment, the iterative version of the Fibonacci sequence is significantly faster beyond when N is greater than 40. When N is less than 40, the recursive and iterative solutions appear to be equally fast and the difference between them is negligible. Beyond this, the recursive solution rate of execution seems to take exponentially longer with each passing iteration. Before N reaches 60, the recursive solution takes over 23 minutes, while the iterative solution continues to execute in under a second.

In terms of Big-O, I would say the recursive solution is exponential, while the iterative solution is likely linear since it is a simple while loop. I think the recursive solution takes so much longer because it calls itself twice within the same return. Each time it must make more variables, allocate more memory and call itself again. This creates a lot of build up of memory and time. The iterative solution doesn't need to keep creating new variables as it overwrites what all ready exists. It is much more direct and so it is faster.

N	Recursive Time (s)	Iterative Time (s)
0	0	0
5	0	0
10	0	0
15	0	0
20	0	0
25	0	0
30	0	0
35	0	0
40	0	0
41	1	0
42	2	0
43	3	0
44	6	0
45	11	0
46	18	0
47	26	0
48	45	0
49	72	0
50	117	0
51	181	0
52	299	0
53	479	0
54	769	0
55	1400	0