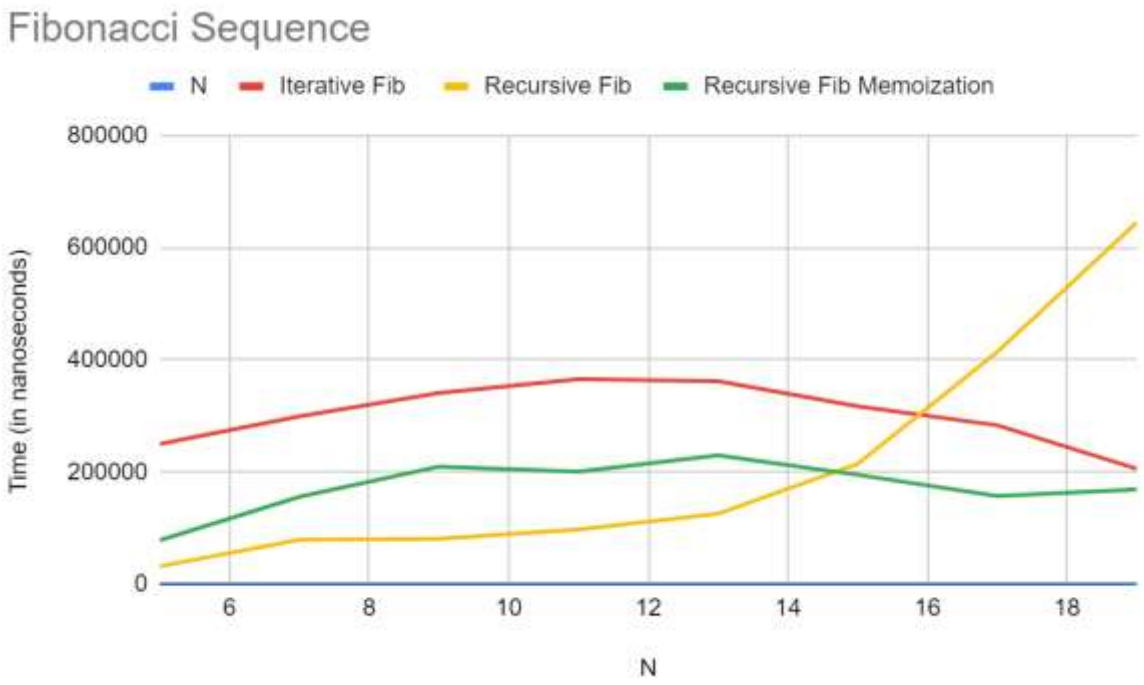
# Fibonacci Sequence Analysis




You can see there are three different functions being measured here. The red line is the iterative implementation of the Fibonacci sequence algorithm. The yellow line is the recursive version and the green line is the recursive function with memorization.

Both the Iterative and the Recursive With Memoization algorithms have time complexity of O(n). However, due to noise in the PC and the very short span of time that is being measured (all times in nanoseconds), the timing diagram became somewhat skewed. I took the average result of multiple runs to get each time, but it still resulted in a skewed graph.

We can see that the recursive Fibonacci algorithm without memorization starts off much faster than the other two but quickly surpasses them. The time complexity of the recursive Fibonacci algorithm ≈ O(2^n).

While the recursive Fibonacci with memorization consistently is faster than the iterative version, it also takes up roughly double the memory of the iterative implementation.