Recursion is an inefficient method of solving Fibonacci number. The approach uses an exponential time complexity in the worst case scenario.When using recursion method to solve Fibonacci number each step forward in the series leads to a more complex problem. This is mainly because a Fibonacci is formed by adding the two preceding numbers.

According to section 3.2 of Niklaus Wirth's excellent book Algorithms and Data Structures:, or its predecessor, Algorithms + Data Structures = Programs: it is best to avoid the use of recursion when there is an obvious solution by iteration. This, however, should not lead to shying away from recursion at any price.

Iteration is a better way of finding Fibonacci number.

Another obvious problem is the fact that it will take much longer for a recursion function to give results for higher numbers like 60, 80 and so on. It has a linear space complexity because there is a heavy push-pop of the stack memory in each recursive call.