Why is it a bad idea to use recursion method to find the Fibonacci of a number?

Scalability and Latency are important factors that should be considered when deciding the choice process for any optimization. The Fibonacci numbers are a sequence of integers in which the first two elements are 0 & 1, and each following element are the sum of the two preceding elements: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ..., 233.

Recursion is a repetitive process in which a function calls itself. It runs in memory as opposed to running against files. Hence, using recursive solutions for find the Fibonacci of a number may involve extensive overhead because they use function calls. Each function call requires push of return memory address, parameters, returned result, etc. and every function return requires that many pops. Each time a function is called, a portion of the memory allocation may be in stack or heap, and if there are large number of recursive calls, running out of memory becomes inevitable.