For the total of the numbers, both the recursive and the iterative go through all the elements, so they both require the same amount of time. The only difference is that recursion is more memory-intensive since the function calls are repeated, but the loop version conserves memory.

For Fibonacci, recursion is slower with large numbers since it does lots of redundant work and is memory-intensive. The loop one is quicker and requires just a few variables, so it is superior.

Conclusion:

So recursion makes the code simpler but mostly takes more memory, and in certain situations like Fibonacci, it can be extremely slower. Iterative solutions are faster and memory-friendly and are usually used when dealing with large inputs. In small calculations such as addition and so on, both are okay, but in heavy calculations, I would use loops instead of recursion.