CO 322 Data Structures and Algorithms

Lab 01

In this lab, we consider four different implementations of a function that calculates a given Fibonacci number, using Java and Python. Those implementations use either recursive method or iterative method for the calculation. Our objective in this lab is to measure the runtime of each of the 4 implementations.

Let's consider the plots corresponding to each different programming language. Here, Figure 01 and Figure 02 correspond to Python. Figure 03 and Figure 04 correspond to Java.

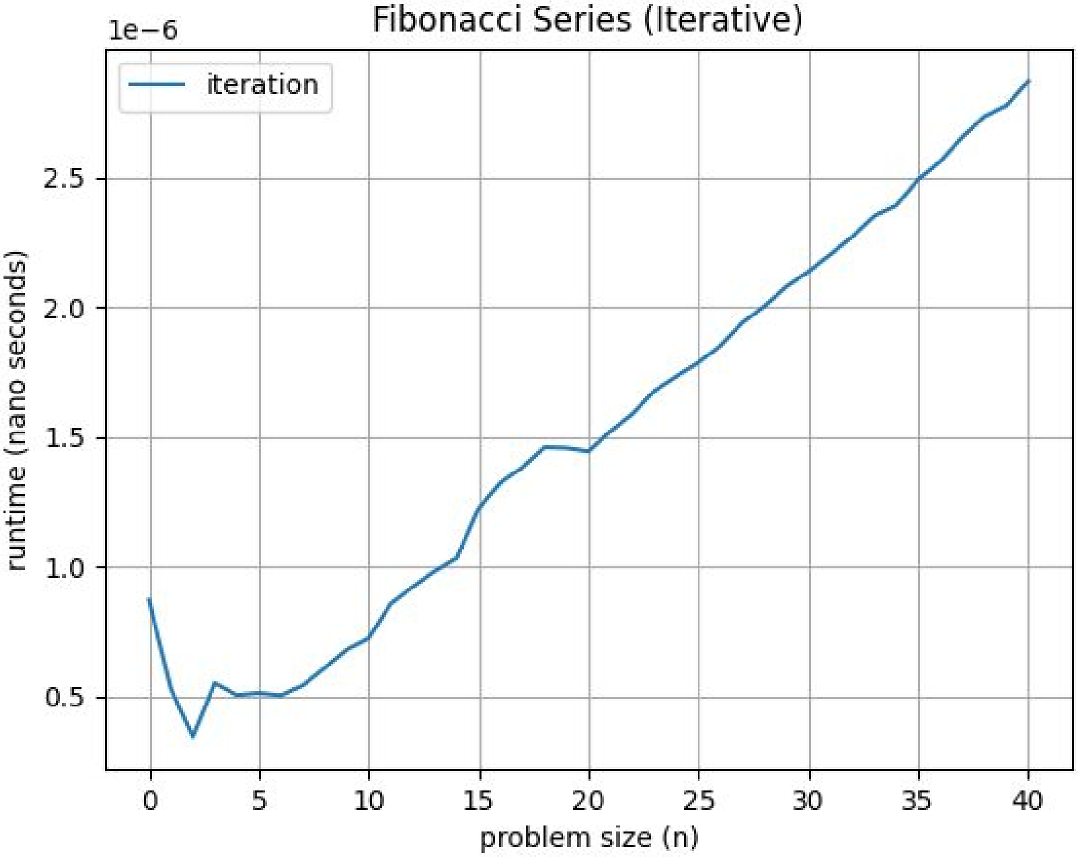

Figure 01

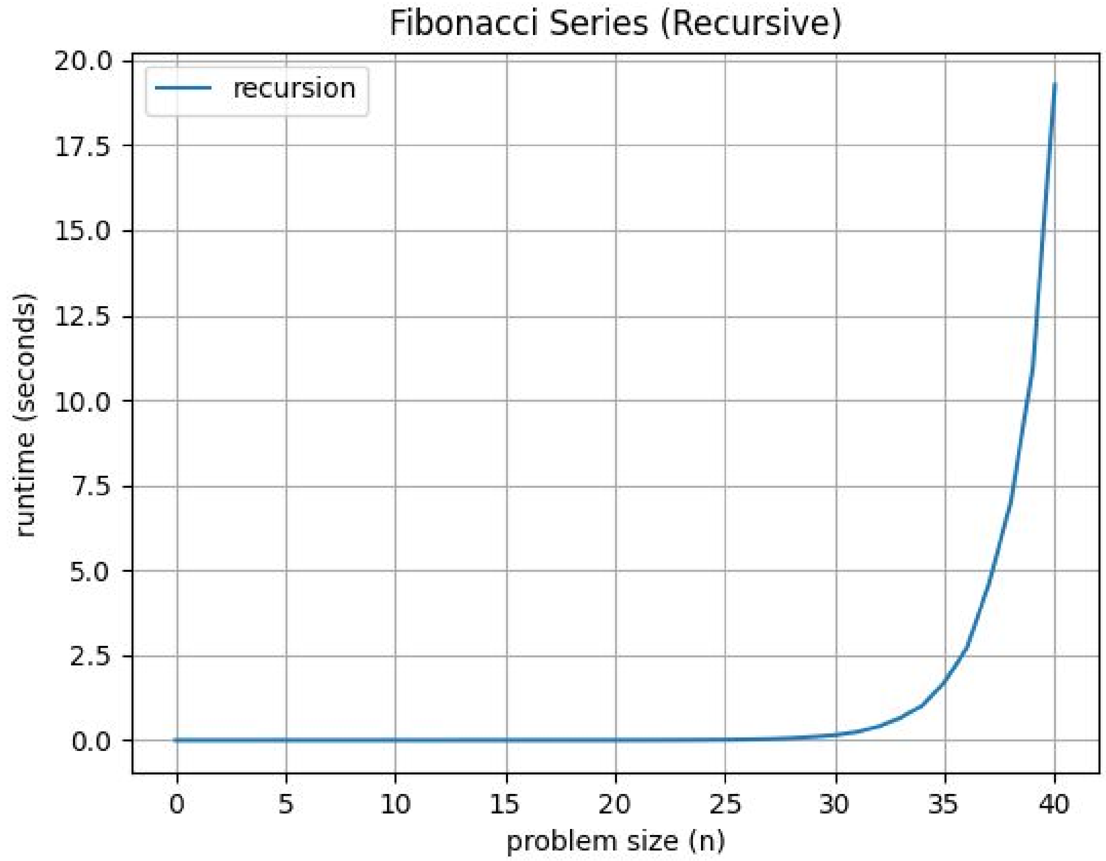

Figure 02

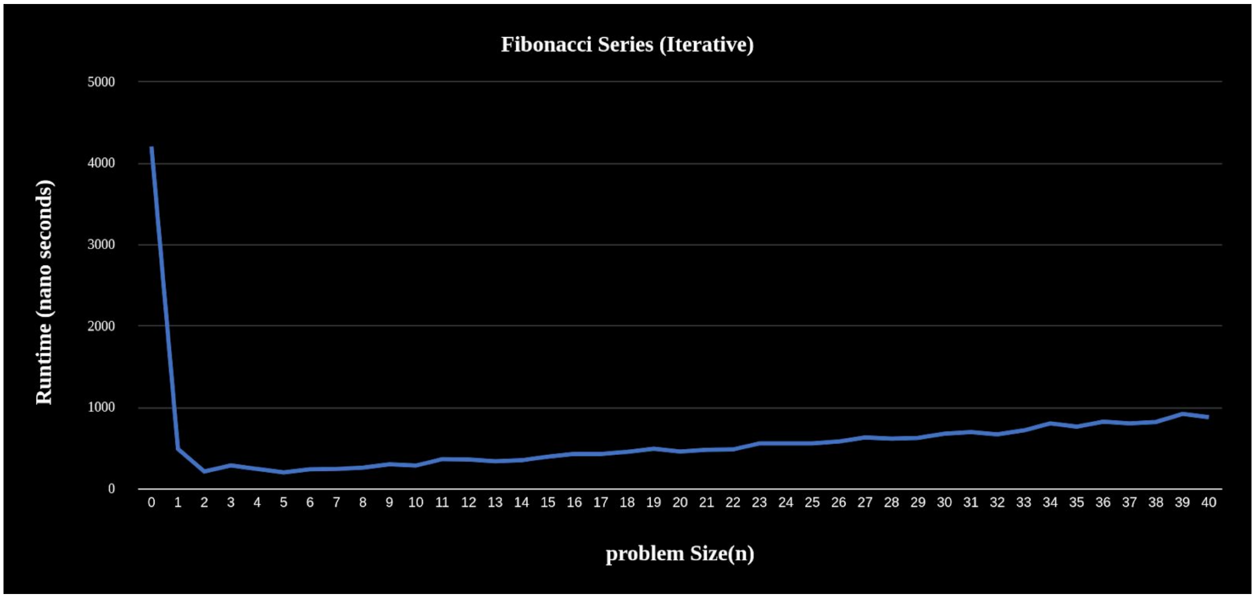

Figure 03

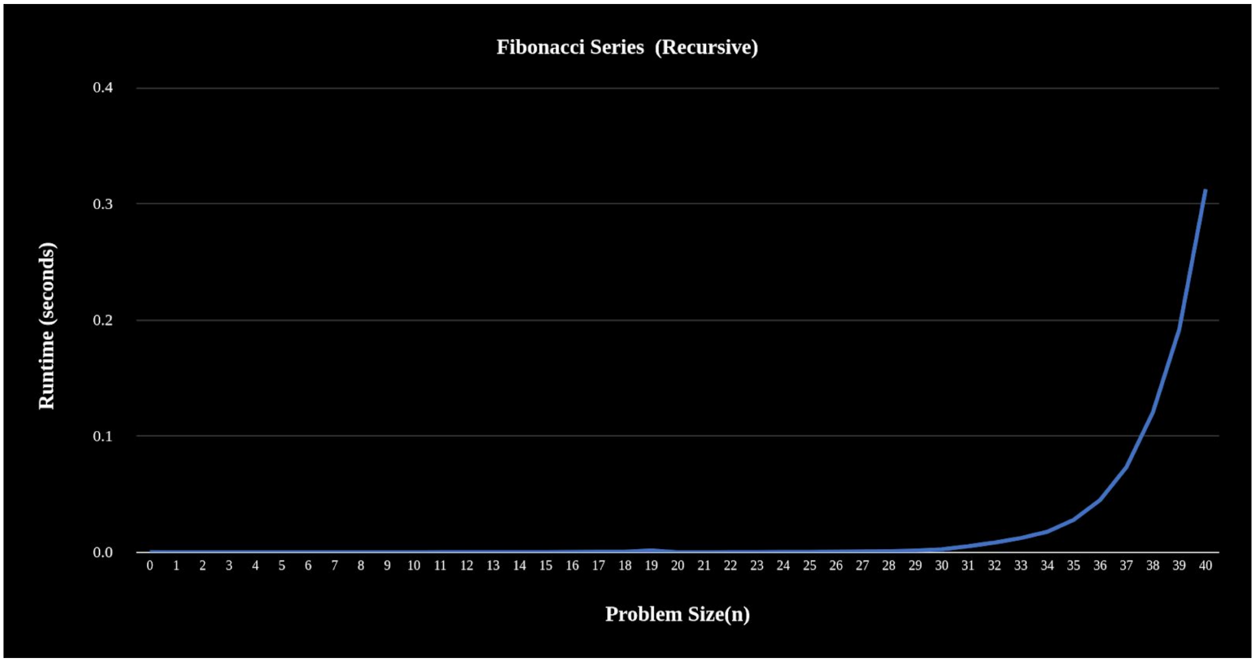

Figure 04

Here if we analyse the resulting runtimes of the two implementations closely, even though for the same programming language, differences in the runtimes could be found. But those **runtime differences get minimized** when the problem size becomes small. This behaviour can be properly examined by plotting both graphs in the same plot. (Figure 05)

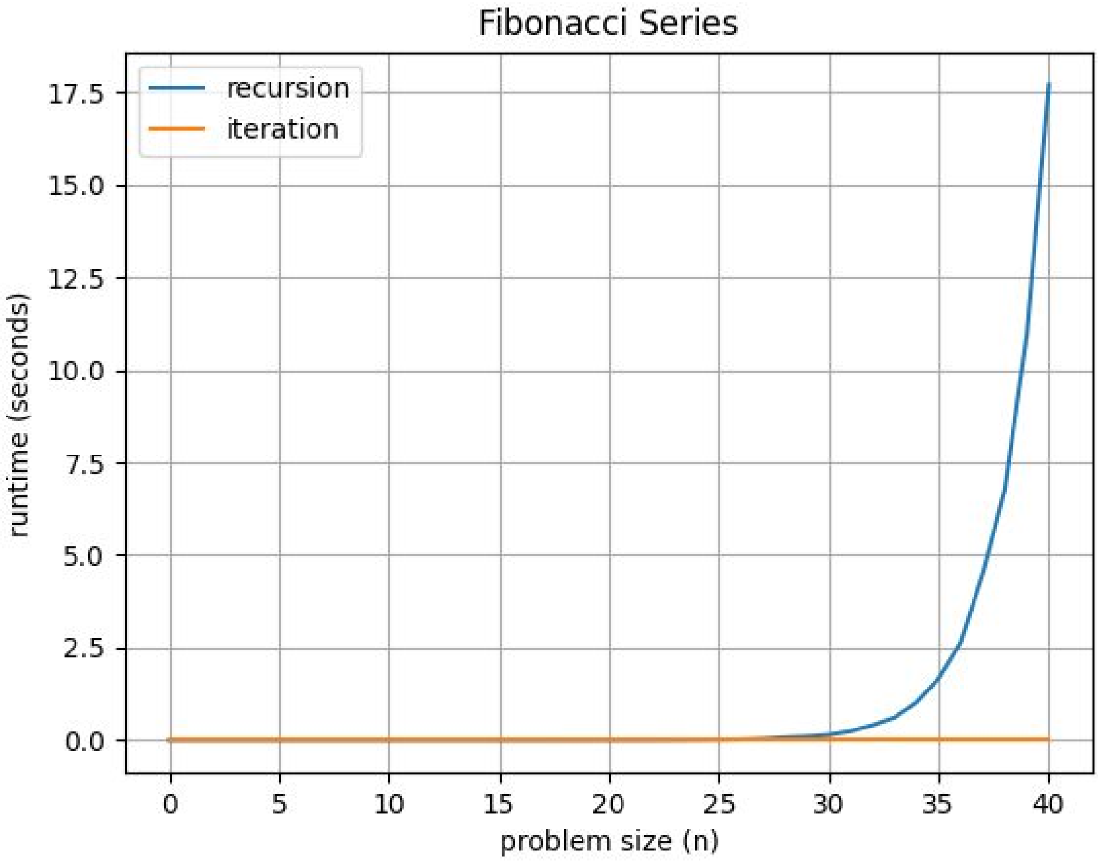

Figure 05

If we analyse the resulting datasets or the plots (Figure 01/02/03/04) that correspond to Java and Python separately, we would find a **difference in the runtime** between the two languages. But the **way the runtime changes is similar** in both languages.

According to Figure 05, **if the problem is small** (n < 30) **both algorithms are useful** and can be used for the calculations. But if the problem is large (n > 30), the recursive method would take more time to execute, compared to the iterative method. Hence for a **large problem** we should avoid using **recursive method** and it **is not useful** when dealing with such scenarios.

We can justify the above mentioned behaviours, by considering the time complexity of each algorithm. Consider the iterative method, which executes n times to return the answer. Hence, the time complexity of the iterative method is O(n). We can analyze the time complexity of a recursive function using the recursive tree. Figure 06 corresponds to the recursion tree for n = 5. Observing the recursion tree, it is obvious that the time complexity of the recursion algorithm is exponential.

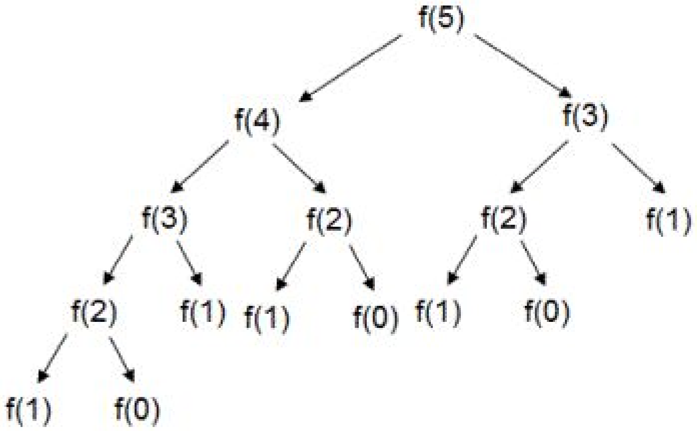

Figure 06

For iterative methods runtime increases gradually as the number of loop iterations increase and for recursive methods runtime increases exponentially as the problem size increases. Hence, this justifies that “If the problem is small both algorithms are useful” and “If the problem is large the recursion algorithm is not useful”.