

CO 322 Data Structures and Algorithms

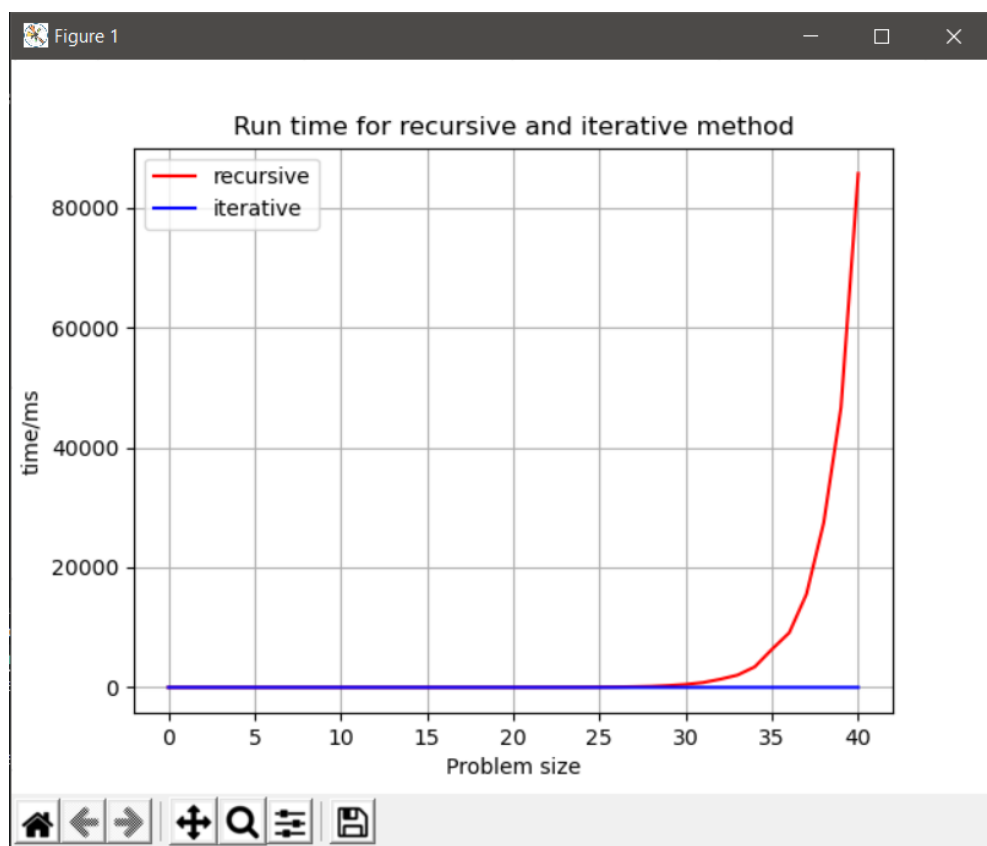
Lab 01

Enumber:E/16/267

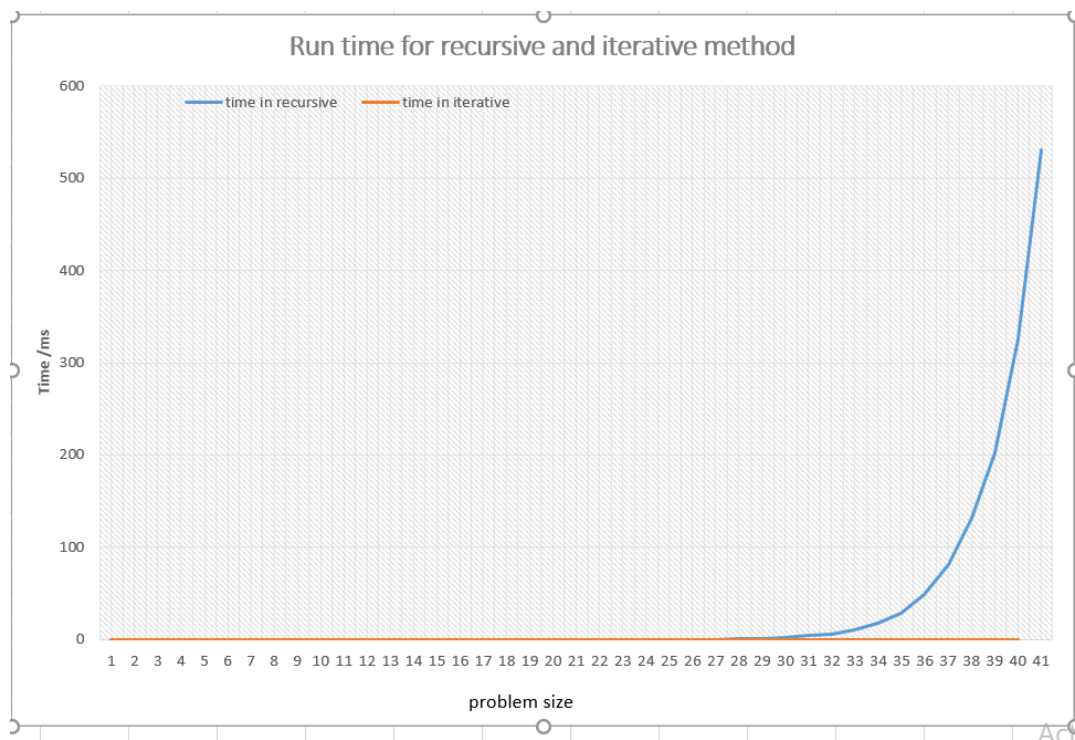
Compare the runtime of two different implementation in two different languages.

In this lab we Compare Runtime of Fibonacci sequence in recursive and iterative method using Java and Python.

Following figures shows plot of runtime of two languages with Problem size.



In python language



In Java language

So in both figures we can see similarities and differences.

When problem size is small there is no a difference in the runtime between the two implementations In the above figures when we consider $n=10$ (small problem size) the runtime of both java and python is equal to 0. When the problem size is small the calculations to be done inside the function will be small in both recursive and iterative methods.

A given algorithm will take different amounts of time on the same inputs depending on such factors as: processor speed; instruction set, disk speed, brand of compiler and etc .In this lab we are checking the run time of two languages JAVA and Python .So when we consider above figures we can see there is a difference in the runtime between the two languages.java is more faster than python in recursive method since the run time of iterative method is 0 in both languages. Java is generally faster and more efficient than Python because it is a compiled language. As an interpreted language, Python has simpler, more concise syntax than Java. It can perform the same function as Java in fewer lines of code.

When we compare the runtime graphs of two languages the shape of two plots is same.so though the time difference differ the way it changes with problem size is not different. Because according Big-O notation the running time of an algorithm for a specific input depends on the number of operations executed. The greater the number of operations, the longer the running time of an algorithm. Though the time taken for that will differ from language to language the way it change will be same for every language.

"If the problem is small both algorithms are useful" yes we can agree for that because when we consider above timing graphs it will prove to us. When problem size is small the runtime is 0 in both algorithms.

The time complexity of the iterative code is linear, as the loop runs from 2 to n. it runs in $O(n)$ time. But The recursive approach seems to be much simpler and smaller, but there is a caveat, as it is calculating the Fibonacci of a number multiple times. Therefore Calculating the time complexity of the recursive approach is not so straightforward.

In recursive method

for $n > 1$:

$T(n) = T(n-1) + T(n-2) + 4$ (1 comparison, 2 subtractions, 1 addition)

time taken by recursive Fibonacci is $O(2^n)$ or exponential. So iterative methods are faster than recursive method. But when the problem size is small the calculations are small in recursive functions. Therefore run time will be same in both algorithms.

"If the problem is large _b r is not useful " ..Yes this is true when we consider above graphs when problem size larger the runtime of recursive method is larger. Run time of the iterative method will remain as 0 when we increase the problem size. Therefore If the problem is large _b r is not useful.