GT Username: sanne31@gatech.edu

PART 1: GET FAMILIAR WITH R

R is a language which is used mostly for statistical computations and graphing various plots. It provides a large multitude of statistical techniques – linear and non-linear modeling techniques such as regression, time series analysis, classification, clustering, etc. – and graphical techniques.

R is also an interpreted language which has several noticeable advantages over the general compiled languages. Some basic differences are:

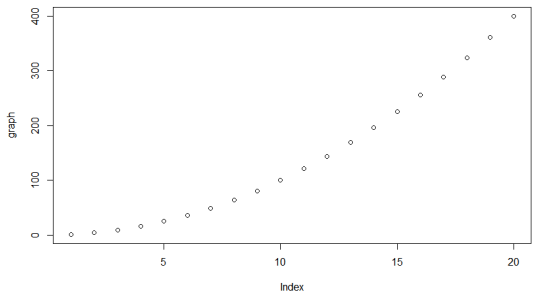
Considering Factor	Compiled Languages	Interpreted Languages
INPUT	Takes an entire program as its input	Takes a single line of code/instruction as its input
OUTPUT	Generates immediate object code	Doesn't generate any immediate object code
SPEED	Executes faster	Executes slower
MEMORY	Requires more memory in order to create object code	Requires less memory as it doesn't create any object code
WORKLOAD	Doesn't need to compile every single time -> only once	Has to convert high level languages to low level programs at the time of execution so inherently the workload is larger than that of compiled languages
ERRORS	Displays errors once the entire program is checked	Displays errors for each line of instruction

A simple example of using R for statistical and graphical analysis:

R code for Graphing $Y = X^2$

```
x = 1:20
y = x*x
plot(y)
```

Plot

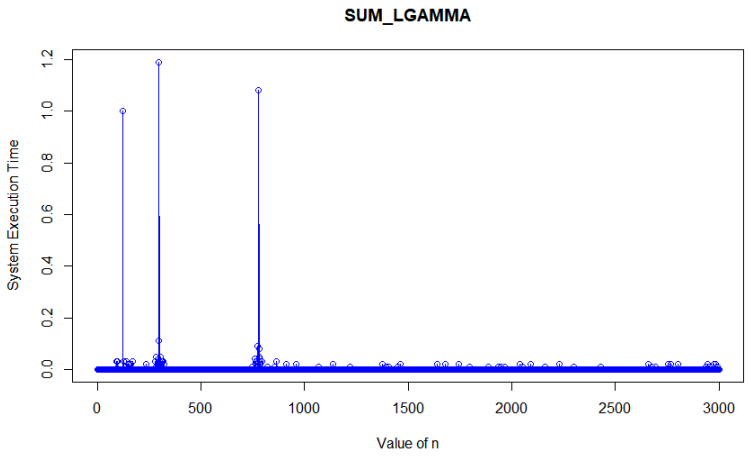


PART 5: COMPARE RESULTS TO BUILT IN R FUNCTION

a. sum of built-in function for calculating log_gamma of n – lgamma(n)

Time complexity of sum_lgamma – O(log(n))

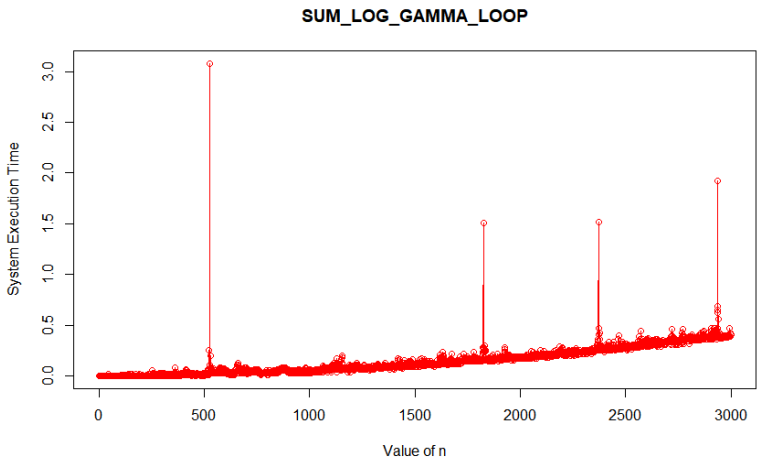
As the value of n increases, the execution time for the built in function to calculate log_gamma remains the same/at a negligible change of rate. From the graph of execution time varying with value of n, we can observe the time complexity to be ***O(log(n))***.



b. sum of the implementation of log_gamma with a loop – log_gamma_loop(n)

Time complexity of sum_log_gamma_loop – O(n)

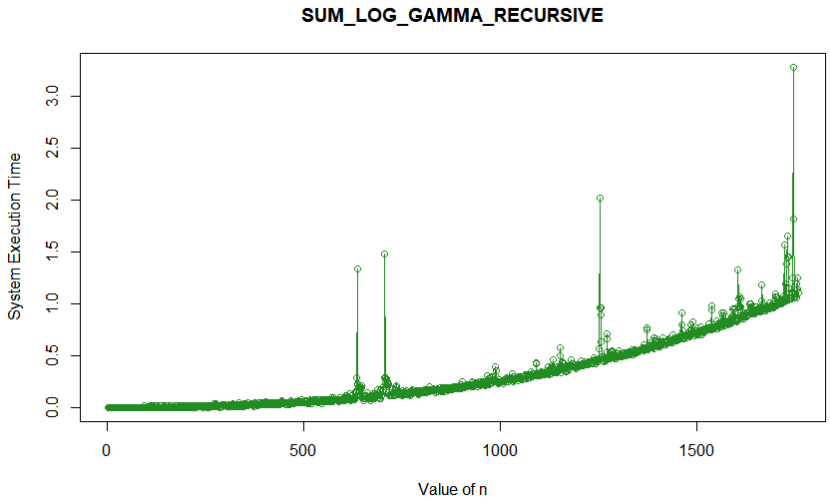
Compared to the built-in function, it is observed that execution time for this particular function is slightly slower than the built-in function. Increases slowly up to a value of n and then increases in the order of n. From the graph below, time complexity is observed to be ***O(n).***



c. sum of implementation of log_gamma using recursion – log_gamma_recursive(n)

Time complexity of sum_log_gamma_recursive – $O(n^2)$

Compared to both functions above, sum_log_gamma_recusive performs much slower than any of the above 2 functions. As the value of n keeps increasing, the system execution time gets squared. Time complexity as observed from the graph is ***O(n^2).***



References: https://www.upwork.com/hiring/development/the-basics-of-compiled-languages-interpreted-languages-and-just-in-time-compilers/ ; http://bigocheatsheet.com/