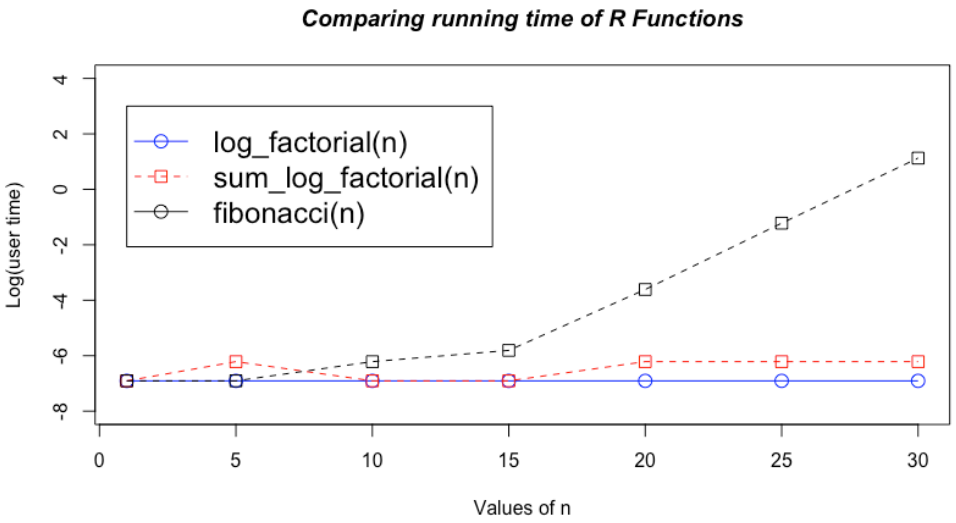

Time complexity of log_factorial(N): O(N*logN); Time complexity of sum_log_factorial(N): $O(N^2*logN)$; Time complexity of fibonacci(N): $O(2^N)$;

```
# Initializing lists for storing running time

time1 <- c()
time2 <- c()
time3 <- c()

# Setting a list of n values

n_values <- seq(0,30,5)
n_values[1] <- 1

# Calculating running time for each functions

for (n in n_values){
    runningTime <- system.time(log_factorial(n))
    userTime <- log(as.numeric(runningTime[1])+0.001)
    time1 <- c(time1, userTime)

    runningTime <- system.time(sum_log_factorial(n))
    userTime <- log(as.numeric(runningTime[1])+0.001)
    time2 <- c(time2, userTime)

    runningTime <- system.time(fibonacci(n))
    userTime <- log(as.numeric(runningTime[1])+0.001)
    time3 <- c(time3, userTime)
}

plot(n_values, time1, cex=1.5, type="o", col="blue", ylim=c(-8,4), ann=FALSE)
lines(n_values, time2, cex=1.5, type="o", pch=22, lty=2, col="red")
lines(n_values, time3, type="o", cex=1.5, pch=22, lty=2, col="black")
title(main="Comparing running time of R Functions", cex=5, col.main="black",
font.main=4)
title(xlab="Values of n", cex=5, col.lab="black")
title(ylab="Log(user time)", cex=5, col.lab="black")
legend(1, 3, c("log_factorial(n)","sum_log_factorial(n)", "fibonacci(n)"), cex=1.5,
col=c("blue","red","black"), pch=21:22, lty=1:2)
```