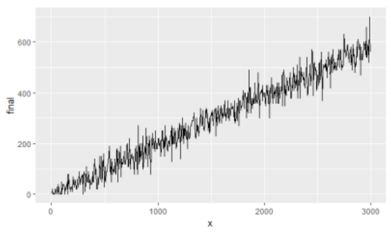
GT ID: 903295488

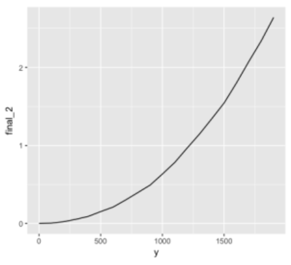
This is the code snippet that I used to computer and plot run times:

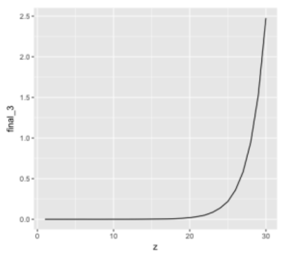
```r
N=3000
x <- seq(1, N, 5)
df <- matrix(ncol=20, nrow=length(x))
for(j in 1:20){
  log_factorialtime <- array()
  for (i in 1:length(x)){
    log_factorialtime[i]=system.time(log_factorial(x[i]))*1000
  }
  df[,j] <- log_factorialtime
}
df <- data.frame(df)
final <- rowSums(df)
df_all <- data.frame(x, final)
p <- ggplot(df_all,aes(x=x, y=final))+geom_line()
N=2000
y <- seq(1, N, 100)
df_2 <- matrix(ncol=20, nrow=length(y))
for(m in 1:20){
  sum_log_factorialtime <- array()
  for (n in 1:length(y)){
    sum_log_factorialtime[n]=system.time(sum_log_factorial(y[n]))
  }
  df_2[,m] <- sum_log_factorialtime
}
df_2 <- data.frame(df_2)
final_2 <- rowMeans(df_2)
df_all_2 <- data.frame(y, final_2)
p <- ggplot(df_all_2,aes(x=y, y=final_2))+geom_line()
N=30
z<- seq(1, N, 1)
df_3 <- matrix(ncol=20, nrow=length(z))
for(m in 1:20){
  fibonaccitime <- array()
  for (n in 1:length(z)){
    fibonaccitime[n]=system.time(fibonacci(z[n]))
  }
  df_3[,m] <- fibonaccitime
}
df_3 <- data.frame(df_3)
final_3 <- rowMeans(df_3)
df_all_3 <- data.frame(z, final_3)
p <- ggplot(df_all_3,aes(x=z, y=final_3))+geom_line()
```

These are 3 separate plots showing running time vs n for the 3 functions and the time complexity in Big-O notation for each plot.

log_factorial: O (n); sum_log_factorial: O (n^2); fibonacci: O (2^n)


final
x


final_2
y


final_3
z