Fibonacci Comparison

Gerhard Bilek

8 3 2019

Goal

Comparison of an inefficient to an efficient type of Fibonacci function. The inefficient type uses only a bare recursive function to generate the Fibonacci numbers, whereas the efficient one has an mamorization tool implemented.

Efficient VS Inefficient

Time Assessment: Command Line Method

```
library(ggplot2)
library(reshape2)

fibo_test <- c(5,10,15,20,25,30,35,5,10,15,20,25,30,35)

efficient <- c(0.04, 0.05, 0.04, 0.05, 0.04, 0.05, 0.05)
inefficient <- c(0.05,0.05,0.05,0.06,0.17,1.25,13.5)

dfx <- data.frame(efficient,inefficient)
dfxm <- melt(dfx)
```

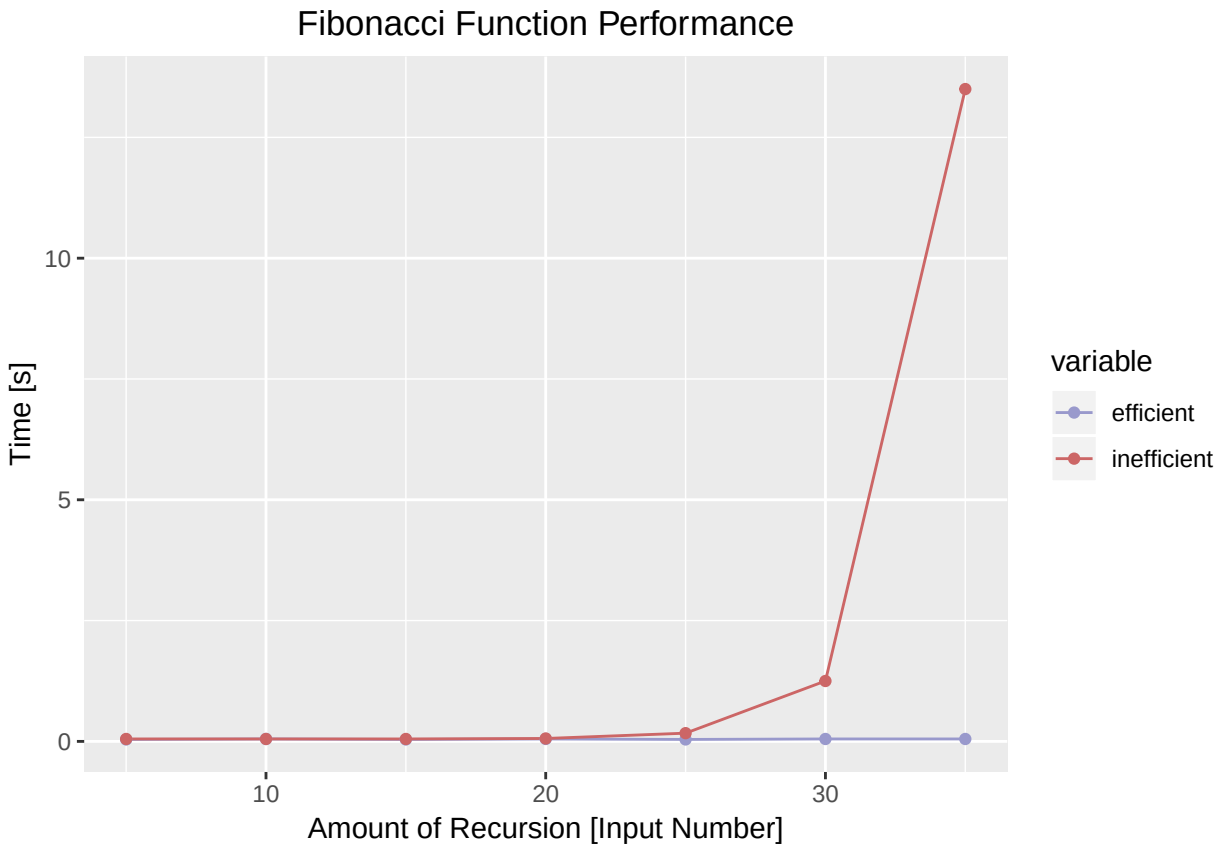
```
## No id variables; using all as measure variables
```

```
df <- cbind(fibo_test,dfxm)
df
```

```
##    fibo_test    variable value
## 1          5   efficient  0.04
## 2         10   efficient  0.05
## 3         15   efficient  0.04
## 4         20   efficient  0.05
## 5         25   efficient  0.04
## 6         30   efficient  0.05
## 7         35   efficient  0.05
## 8          5 inefficient  0.05
## 9         10 inefficient  0.05
## 10        15 inefficient  0.05
## 11        20 inefficient  0.06
## 12        25 inefficient  0.17
## 13        30 inefficient  1.25
## 14        35 inefficient 13.50
```

```
ggplot(df, aes(x=fibo_test, y=value, group=variable, color=variable))+geom_line()+
  geom_point()+ggtitle("Fibonacci Function Performance")+
  xlab("Amount of Recursion [Input Number]")+
  ylab("Time [s]")+
```

```
  theme(plot.title = element_text(hjust = 0.5))+
  scale_color_manual(values=c("#9999CC", "#CC6666"))
```



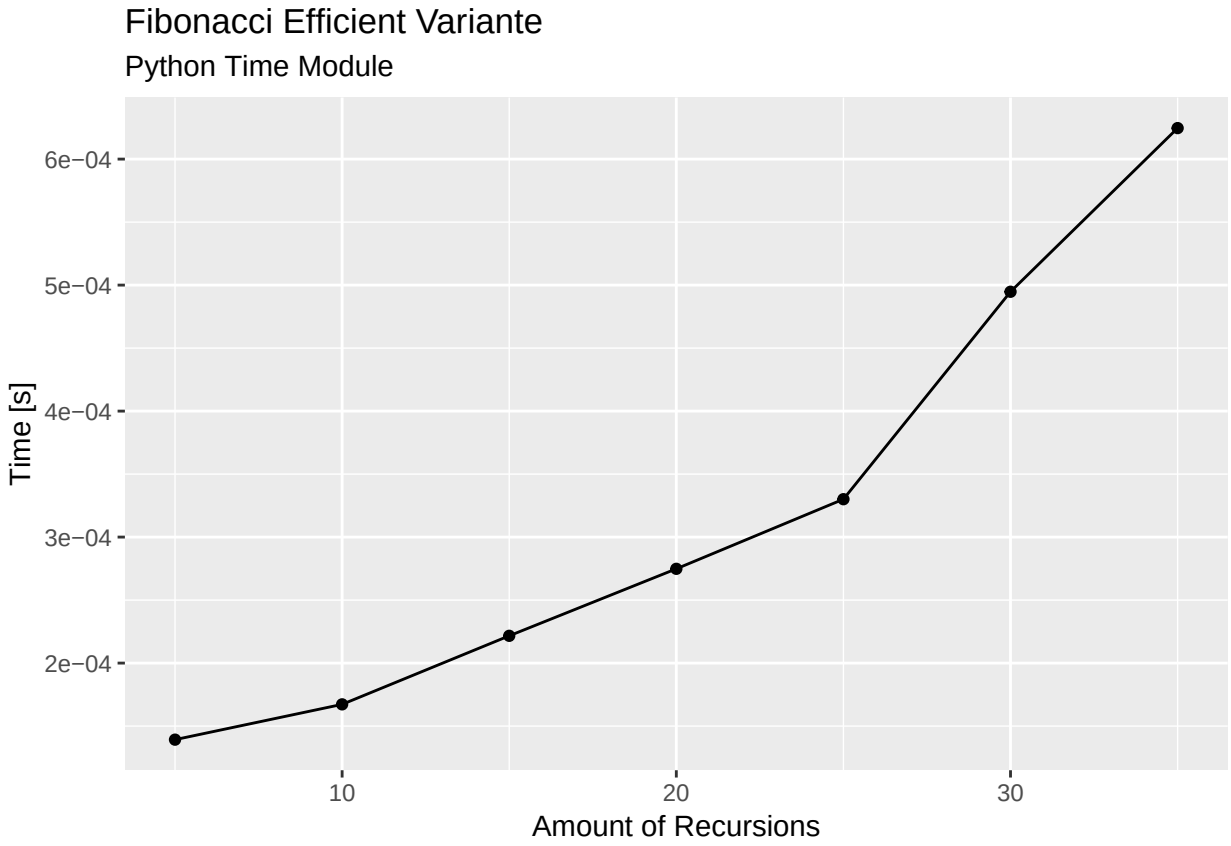
The inefficient function shows an exponential increase in time. The efficient function has a linear increase of its time consume. However, the linear increase could not be detected within the selected amount of recursion using the command line mdoule. Thus the following section shows duration data which were recorded by an Phython time module.

Time Assessment: Python Time Module

```
df_p <- data.frame(numbers=c(5,10,15,20,25,30,35), time=c(0.0001392240000000003, 0.0001672370000000064
df_p
```

```
##   numbers        time
## 1       5 0.000139224
## 2      10 0.000167237
## 3      15 0.000221642
## 4      20 0.000274817
## 5      25 0.000330061
## 6      30 0.000494719
## 7      35 0.000624606
```

```
ggplot(data=df_p, aes(x=numbers, y=time, group=1)) +
  geom_line()+
  geom_point()+labs(title = "Fibonacci Efficient Variante", subtitle = "Python Time Module") + xlab("Am
```



Already for lower amounts of recursive calls, the Python time modle allows for a closer look on the time performance of the efficient Fibonacci function. A linear relation can be shown.