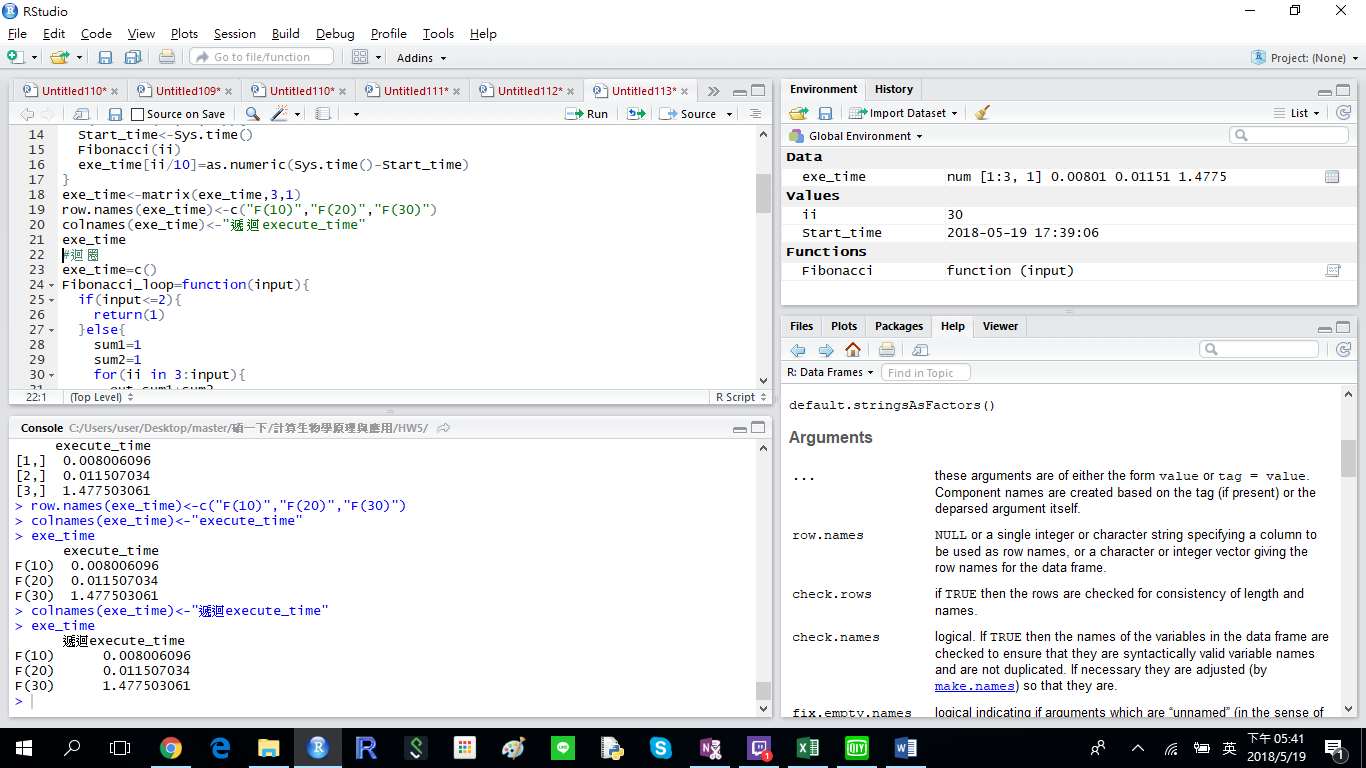
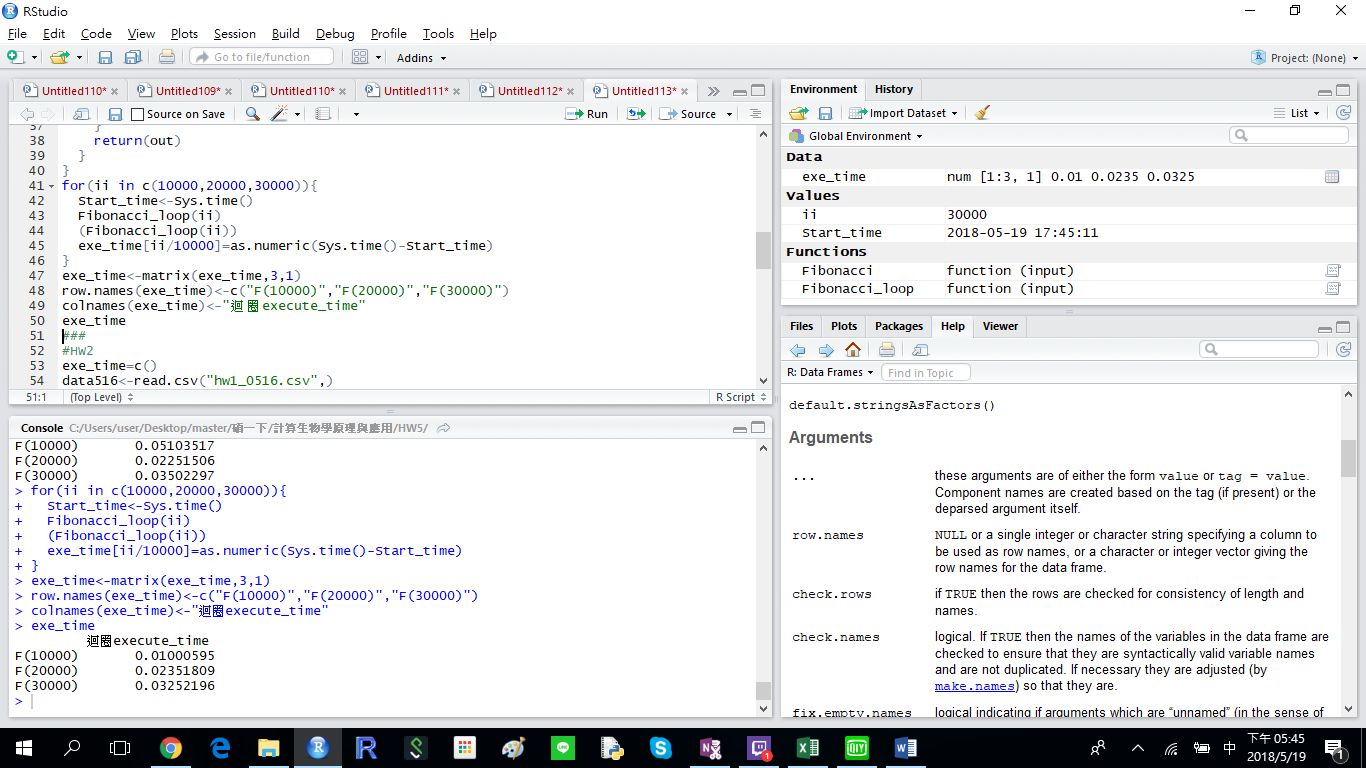
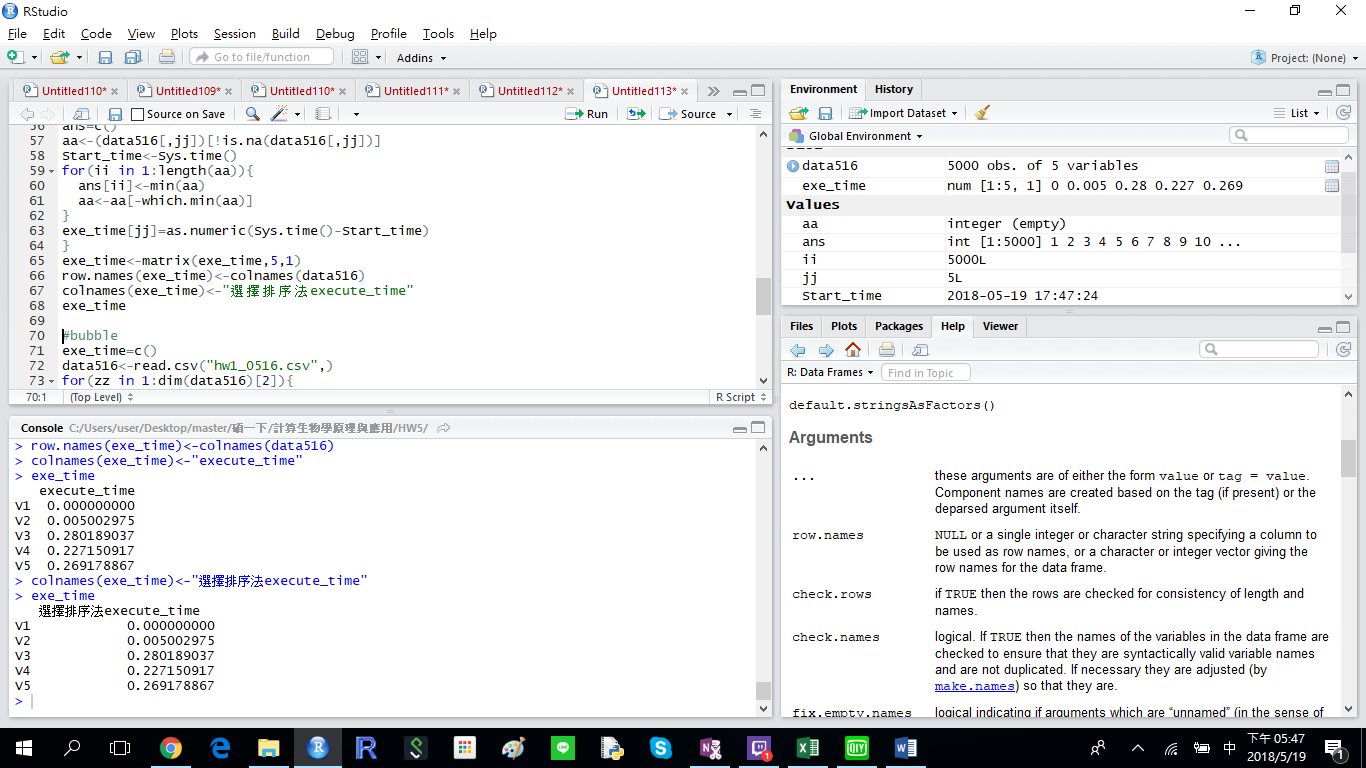
* 請用之前撰寫的遞迴費氏數列及迴圈費 式數列程式碼，分別跑F(10), F(20), F(30)。 請用程式紀錄跑的時間。
* 遞迴執行時間:



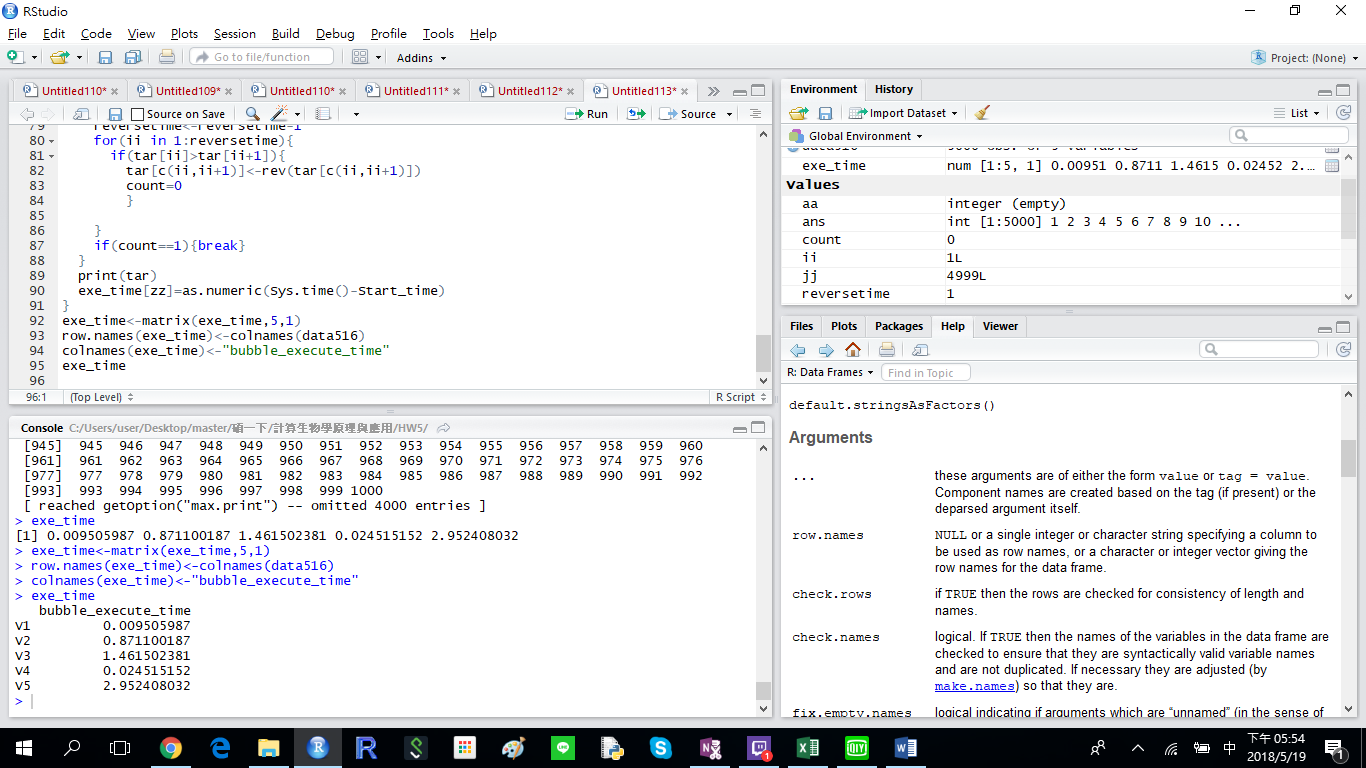
* 迴圈執行時間



* 請讀入hw1\_0516.csv檔案，並針對V1, V2, V3,V4,V5使用這兩個函數進行排序， 並用程式紀錄跑的時間，比較一下兩個 方法在input不同時候有甚麼差異。
* 選擇排序執行時間



* Bubble 執行時間



Bubble表現在大部分時間都表現得比選擇排序法差，除了好的情況(資料排序為從最小排到最大)時，會花費較少時間。

#Rcode

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-

rm(list = ls())

setwd("C:/Users/user/Desktop/master/碩一下/計算生物學原理與應用/HW5")

#遞迴

#HW1

exe\_time=c()

Fibonacci=function(input){

if(input<=2){

return(1)

}else{

return(Fibonacci(input-1)+Fibonacci(input-2))

}

}

for(ii in c(10,20,30)){

Start\_time<-Sys.time()

Fibonacci(ii)

exe\_time[ii/10]=as.numeric(Sys.time()-Start\_time)

}

exe\_time<-matrix(exe\_time,3,1)

row.names(exe\_time)<-c("F(10)","F(20)","F(30)")

colnames(exe\_time)<-"遞迴execute\_time"

exe\_time

#迴圈

exe\_time=c()

Fibonacci\_loop=function(input){

if(input<=2){

return(1)

}else{

sum1=1

sum2=1

for(ii in 3:input){

out=sum1+sum2

if(ii%%2==0){

sum1=out

}else{

sum2=out

}

}

return(out)

}

}

for(ii in c(10000,20000,30000)){

Start\_time<-Sys.time()

Fibonacci\_loop(ii)

(Fibonacci\_loop(ii))

exe\_time[ii/10000]=as.numeric(Sys.time()-Start\_time)

}

exe\_time<-matrix(exe\_time,3,1)

row.names(exe\_time)<-c("F(10000)","F(20000)","F(30000)")

colnames(exe\_time)<-"迴圈execute\_time"

exe\_time

###

#HW2

exe\_time=c()

data516<-read.csv("hw1\_0516.csv",)

for(jj in 1:dim(data516)[2]){

ans=c()

aa<-(data516[,jj])[!is.na(data516[,jj])]

Start\_time<-Sys.time()

for(ii in 1:length(aa)){

ans[ii]<-min(aa)

aa<-aa[-which.min(aa)]

}

exe\_time[jj]=as.numeric(Sys.time()-Start\_time)

}

exe\_time<-matrix(exe\_time,5,1)

row.names(exe\_time)<-colnames(data516)

colnames(exe\_time)<-"選擇排序法execute\_time"

exe\_time

#bubble

exe\_time=c()

data516<-read.csv("hw1\_0516.csv",)

for(zz in 1:dim(data516)[2]){

tar<-(data516[,zz])[!is.na(data516[,zz])]

reversetime<-length(tar)

Start\_time<-Sys.time()

for(jj in 1:(length(tar)-1)){

count=1

reversetime<-reversetime-1

for(ii in 1:reversetime){

if(tar[ii]>tar[ii+1]){

tar[c(ii,ii+1)]<-rev(tar[c(ii,ii+1)])

count=0

}

}

if(count==1){break}

}

print(tar)

exe\_time[zz]=as.numeric(Sys.time()-Start\_time)

}

exe\_time<-matrix(exe\_time,5,1)

row.names(exe\_time)<-colnames(data516)

colnames(exe\_time)<-"bubble\_execute\_time"

exe\_time