Carson Slater

6375 Homework 1 Solutions

```
set.seed(613)
```

Question 1

I read the content on the complex numbers.

Affirmative.

Question 2

I read Chapter 1 of Trefethen and Bau.

Affirmative.

Question 3

```
total <- function(x, verbose = FALSE) {
  total <-  0
  for (i in 1:length(x)) {
    if (verbose == TRUE) {print(total)}
    total <- total + x[i]
  }
  total
}
```

```
# My function
total(1:3, verbose = FALSE)
# [1] 6
```

Question 4

```
dot <- function(x, y, verbose = FALSE) {
  total <- 0
  if (length(x) != length(y)) {
    stop("Dimensions must be equal between vectors.")
    }
  else{
    for (i in 1:length(x)) {
      if (verbose == TRUE) {print(total)}
      total <- total + x[i]*y[i]
    }
  }
  total
}
```

```
x <- c(1:3)
y <- c(4:6)

# My function
dot(x, y, verbose = FALSE)
# [1] 32
```

Question 5

```
mat.vec.prod <- function(A, x) {
  b <- vector()
  for (i in 1:(nrow(A))) {
    b[i] <- dot(x, A[i,])
  }
  b
}
```

```
A <- matrix(1:12, nrow = 3, ncol = 4)
x <- 1:4

# My function
mat.vec.prod(A, x)
# [1] 70 80 90
```

Question 6

```
mat.mat.prod <- function(A, B) {
  if (is.matrix(A) == FALSE | is.matrix(B) == FALSE){
    stop("This function only takes matrices as parameters.")
  }
  if (ncol(A) != nrow(B)) {
    stop("Dimensions for matrices do not match.")
  }
  C <- matrix(nrow = nrow(A), ncol = ncol(B))
  for (i in 1:nrow(A)) {
    for (j in 1:ncol(B)) {
      C[i, j] <- dot(A[i,], B[,j])
    }
  }
  C
}
```

```
A <- matrix(1:12, nrow = 3, ncol = 4)
B <- matrix( 1:8, nrow = 4, ncol = 2)

# My function
mat.mat.prod(A, B)
#      [,1] [,2]
# [1,]   70  158
# [2,]   80  184
# [3,]   90  210
```

Question 7

Since a microsecond equals 1,000 nanoseconds, we can say the median time for the base `R` function was approximately 445 times faster than my function. The median time for the base `R` function was 328.1 nanoseconds, while the median time for my function was 14.6 microseconds. (This was on my 10-core M1 Macbook Pro, my iMac at the department actually took almost 3 times as long for run time in both functions.)

```
bnch <- bench::mark(
  A %*% B,
  mat.mat.prod(A, B)
)
```

```
bnch
# # A tibble: 2 x 6
#   expression             min   median `itr/sec` mem_alloc `gc/sec`
#   <bch:expr>         <bch:tm> <bch:tm>     <dbl> <bch:byt>    <dbl>
# 1 A %*% B              721ns    826ns   949831.        0B      0
# 2 mat.mat.prod(A, B)    27us   28.9us    32789.        0B     19.7
```

Question 8

- $((AB)C)D$ would take 312500 operations.
- $(AB)(CD)$ would take 1.3e+07 operations.
- $(A(BC))D$ would take 56000 operations.
- $A((BC)D)$ would take 1025000 operations.
- $A(B(CD))$ would take 6250000 operations.

```
# Benchmarking the five associative combinations.
matrix_bench <- bench::mark(
  ((A %*% B) %*% C) %*% D,
  (A %*% B) %*% (C %*% D),
  A %*% (B %*% C) %*% D,
  A %*% ((B %*% C) %*% D),
  A %*% (B %*% (C %*% D))
)
matrix_bench
# # A tibble: 5 x 6
#   expression                   min   median `itr/sec` mem_alloc `gc/sec`
#   <bch:expr>                <bch:tm> <bch:tm>     <dbl> <bch:byt>    <dbl>
# 1 ((A %*% B) %*% C) %*% D 221.56us 433.41us     2204.  488.81KB    21.3
# 2 (A %*% B) %*% (C %*% D)   8.38ms   9.55ms      105.    2.38MB     4.19
# 3 A %*% (B %*% C) %*% D    59.07us 214.35us     4898.  391.31KB    39.8
# 4 A %*% ((B %*% C) %*% D) 649.91us 902.31us     1099.  547.17KB    10.7
# 5 A %*% (B %*% (C %*% D))   4.21ms   5.22ms      197.    2.44MB    11.2
```