hw1_37810_weidi

Section 4.1.1

Exercise 2

(a)

```r
Fibonacci <- numeric(30)
Fibonacci[1] <- Fibonacci[2] <- 1
for (i in 3:30) Fibonacci[i] <- Fibonacci[i - 2] + Fibonacci[i - 1]
print(Fibonacci)
```

```
##  [1]      1      1      2      3      5      8     13     21     34     55
## [11]     89    144    233    377    610    987   1597   2584   4181   6765
## [21]  10946  17711  28657  46368  75025 121393 196418 317811 514229 832040
```

```r
#Compute f(n)/f(n-1)

print(Fibonacci[2:30]/Fibonacci[1:29])
```

```
##  [1] 1.000000 2.000000 1.500000 1.666667 1.600000 1.625000 1.615385
##  [8] 1.619048 1.617647 1.618182 1.617978 1.618056 1.618026 1.618037
## [15] 1.618033 1.618034 1.618034 1.618034 1.618034 1.618034 1.618034
## [22] 1.618034 1.618034 1.618034 1.618034 1.618034 1.618034 1.618034
## [29] 1.618034
```

Yes, the sequence appears to be converging to 1.618034.

(b)

```r
(1+sqrt(5))/2
```

```
## [1] 1.618034
```

Yes, the sequence is converging to this ratio. Proof: f(n) = f(n-2) + f(n-1) f(n)/f(n-1) = f(n-2)/f(n-1) + 1 limf(n)/f(n-1) = limf(n-2)/f(n-1) + 1 L = 1/L + 1 L^2 - L - 1 = 0 L = (1+sqrt(5))/2

Exercise 3

(a)

The final value of answer should be 1+2+3+4+5 = 15

```r
answer <- 0
for (j in 1:5) answer <- answer + j
answer
```

```
## [1] 15
```

(b)

The final value of answer should be c(1,2,3,4,5)

```r
answer <- NULL
for (j in 1:5) answer <- c(answer, j)
answer
```

```
## [1] 1 2 3 4 5
```

(c)

The final value of answer should be c(0,1,2,3,4,5)

```
answer <- 0
for (j in 1:5) answer <- c(answer, j)
answer
```

```
## [1] 0 1 2 3 4 5
```

(d)

The final value of answer should be 1*2*3*4*5 = 120

```
answer <- 1
for (j in 1:5) answer <- answer * j
answer
```

```
## [1] 120
```

(e)

The final value of answer should be 3

```
answer <- 3
for (j in 1:15) answer <- c(answer, (7 * answer[j]) %% 31)
answer
```

```
##  [1]  3 21 23  6 11 15 12 22 30 24 13 29 17 26 27  3
```

Section 4.1.2

Exercise 4

```
# p is the initial investment amount, n is the number of interest periods
compInterest <- function(p,n) {
  if(n<=3) i<-0.04 else i<-0.05
  interest = p*((1+i)^n-1)
  return (interest)
}
```

Exercise 5

```
#function to calculate a monthly mortgage payment R where i is an interest rate (compounded monthly),P
compPayment <- function(n,p,openTerm) {
  if(openTerm==TRUE) i <- 0.005 else i<-0.004
  payment = p*i/(1-(1+i)^(-n))
  return (payment)
}
```

Section 4.1.3

Exercise 2

```
Fibonacci <- c(1, 1)
while (max(Fibonacci) < 300) {
  n = length(Fibonacci)
  Fibonacci <- c(Fibonacci, Fibonacci[n-1]+Fibonacci[n])
}
```

```r
Fibonacci
```

```
##  [1]   1   1   2   3   5   8  13  21  34  55  89 144 233 377
```

Exercise 4

```r
i <- 0.006
i2 <- (1 - (1 + i)^(-20)) / 19

while (abs(i-i2) > 0.000001) {
  i <- i2
  i2 <- (1 - (1 + i2)^(-20)) / 19
}
i2
```

```
## [1] 0.004954139
```

If we try other starting guesses of i, such as i = 0.1

```r
i <- 0.1
i2 <- (1 - (1 + i)^(-20)) / 19

while (abs(i-i2) > 0.000001) {
  i <- i2
  i2 <- (1 - (1 + i2)^(-20)) / 19
}
i2
```

```
## [1] 0.004953499
```

The final values of i are very close regardless of the starting guesses.

Exercise 5

```r
i <- 0.006
i2 <- (1 - (1 + i)^(-20)) / 19
n_iter = 1

while (abs(i-i2) > 0.000001) {
  i <- i2
  i2 <- (1 - (1 + i2)^(-20)) / 19
  n_iter = n_iter + 1
}
i2
```

```
## [1] 0.004954139
```

```r
paste('The number of iterations needed is ',n_iter)
```

```
## [1] "The number of iterations needed is  74"
```

Section 4.1.5

Exercise 2

Once p >= sqrt(n), all the facotrs of the composite numbers have been exhausted, thus all remaining entries in sieve are prime

```r
Eratosthenes <- function(n) {
# Print prime numbers up to n (based on the sieve of Eratosthenes)
if (n >= 2) {
  sieve <- seq(2, n)
  primes <- c()
  while (length(sieve) > 0) {
    p <- sieve[1]
    primes <- c(primes, p)
    sieve <- sieve[(sieve %% p) != 0]
    if(p>=sqrt(n)) return(c(primes,sieve))
  }
  return(primes)

} else {
    stop("Input value of n should be at least 2.")
}
}
```

```r
Eratosthenes(30)
```

```
##  [1]  2  3  5  7 11 13 17 19 23 29
```

Section 4.2.1

Exercise 2

(a)

```r
compound.interest <- function(p,i,n) {
  fv = p*(1+i)^n
  return(fv)
}
```

(b)

```r
#how much will Mr. Ng have in the bank at the end of 30 months, if he deposits $1000, and the interest
compound.interest(1000,0.01,30)
```

```
## [1] 1347.849
```

Exercise 3

```r
bisection <- function(f, x1, x2) {
  repeat{

    x3 <- (x1 + x2)/2

    if(f(x3) == 0) {
      return(x3)
    }

    if (f(x3)*f(x1) > 0) {
      x1 <- x3
    } else {
      x2 <- x3
```

```
    }

    if (abs(x1 - x2) < 1e-15) {
    return((x1 + x2)/2)
    }
  }
}
```

```
bisection(function(x) x^2-4,1,3)
```

```
## [1] 2
```

Section 4.4.1

Exercise 1

```
mergesort <- function (x, decreasing=FALSE) {
  len <- length(x)
  if (len<2) result<-x
  else {
    y <- x[1:(len%/%2)]
    z <- x[(len%/%2+1):len]
    y <- mergesort(y,decreasing)
    z <- mergesort(z,decreasing)
    result <- c()
    while (min(length(y),length(z)) > 0) {
        if ((!decreasing && y[1]<z[1]) || (decreasing && y[1]>z[1])) {
        result <- c(result,y[1])
        y <- y[-1]
        } else {
        result <- c(result,z[1])
        z <- z[-1]
        }
    }


  if (length(y) > 0)
  result <- c(result,y)
  else
  result <- c(result,z)

  }
return(result)
}
```

```
mergesort(c(1,4,2,7,3,5),TRUE)
```

```
## [1] 7 5 4 3 2 1
```

```
mergesort(c(1,4,2,7,3,5))
```

```
## [1] 1 2 3 4 5 7
```

Section 4.4.3 (2016 edition, not required)

Exercise 1

```r
factorial(10)
```

```
## [1] 3628800
```

```r
factorial(50)
```

```
## [1] 3.041409e+64
```

```r
factorial(1000)
```

```
## Warning in factorial(1000): value out of range in 'gammafn'
```

```
## [1] Inf
```

Exercise 2

(a)

```r
binom <- function(n,m) {
  return(factorial(n)/(factorial(n-m)*factorial(m)))
}
```

(b)

```r
binom(4,2)
```

```
## [1] 6
```

```r
binom(50,20)
```

```
## [1] 4.712921e+13
```

```r
binom(5000,2000)
```

```
## Warning in factorial(n): value out of range in 'gammafn'
```

```
## Warning in factorial(n - m): value out of range in 'gammafn'
```

```
## Warning in factorial(m): value out of range in 'gammafn'
```

```
## [1] NaN
```

(c)

```r
binom_new <- function(n,m) {
  return(exp(sum(log(1:n))-sum(log(1:(n-m)))-sum(log(1:m))))
}
```

(d)

```r
binom_new(4,2)
```

```
## [1] 6
```

```r
binom_new(50,20)
```

```
## [1] 4.712921e+13
```

```r
binom_new(5000,2000)
```

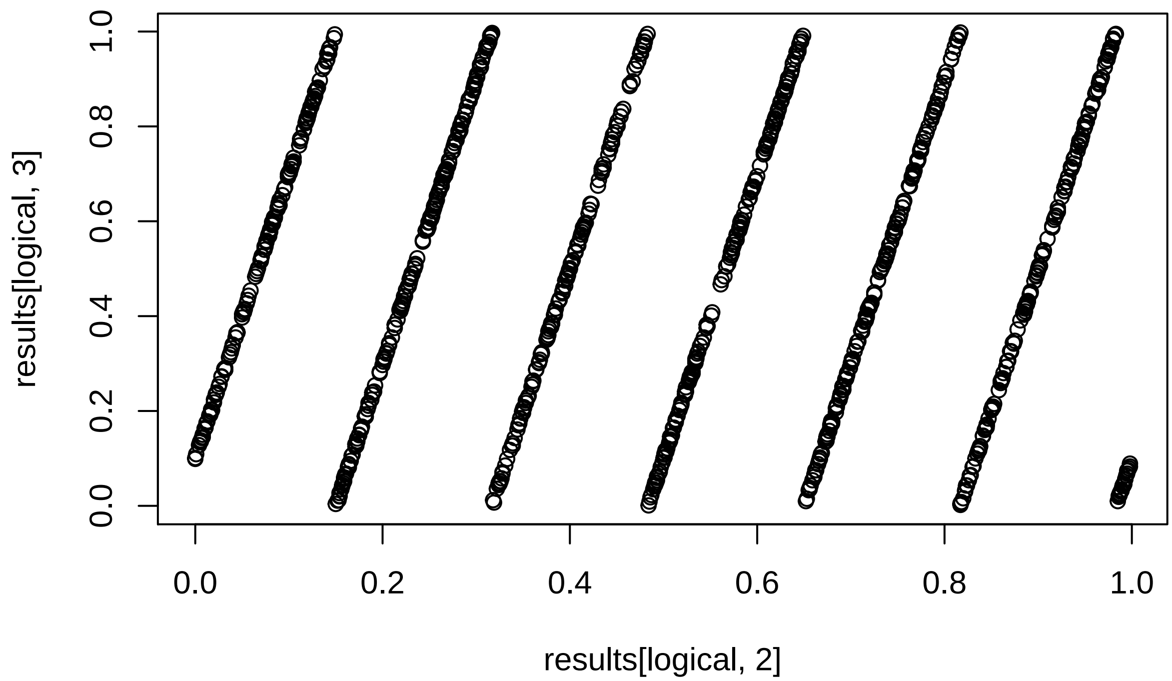
```
## [1] Inf
```

Chapter 4 (2016 edition)

Exercise 1

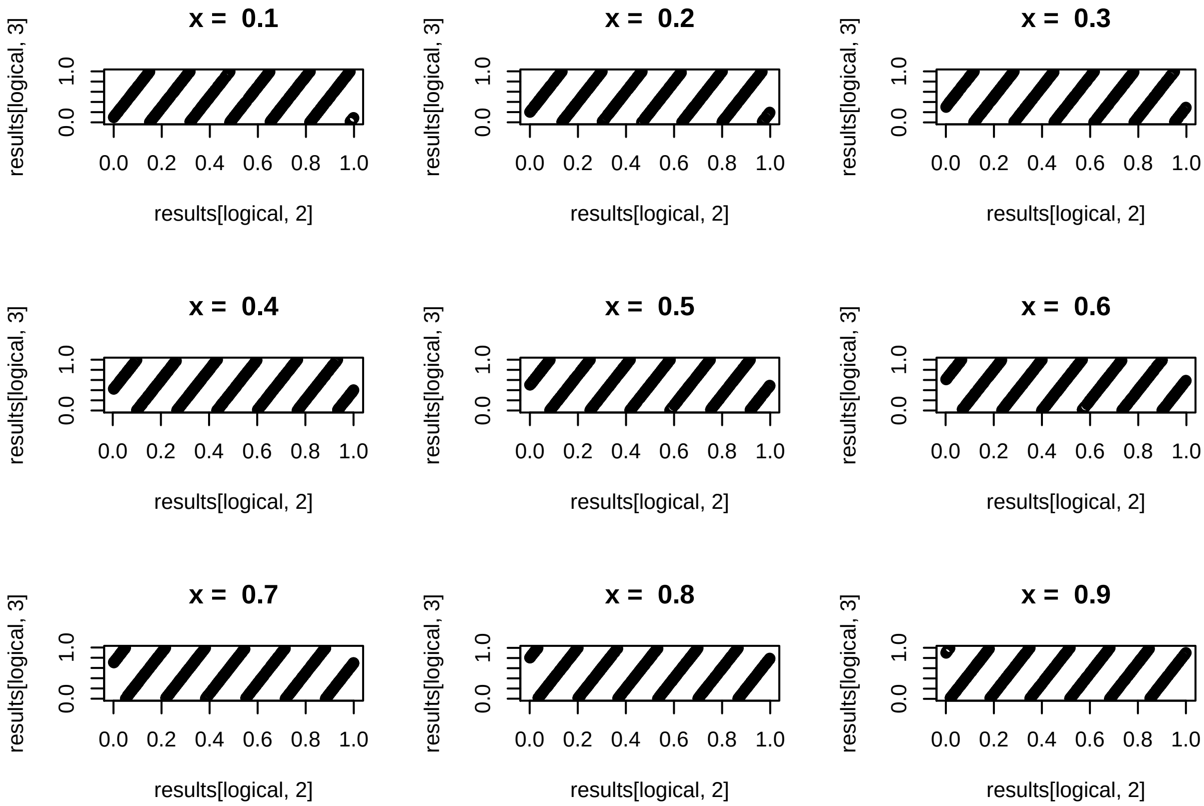
```
results <- numeric(3000000)
x4 <- 123
for (i in 1:3000000) {
x4 <- (65539*x4) %% (2^31)
results[i] <- x4 / (2^31)
}
results <- matrix(round(results, 3), ncol = 3, byrow = TRUE)

logical<-results[,1]==0.1
plot(results[logical,2],results[logical,3])
```




```
par(mfrow = c(3, 3))

for (i in seq(0.1,0.9,0.1)) {
  logical<-results[,1]==round(i,3)
  plot(results[logical,2],results[logical,3],main=paste("x = ",i))
}
```



Exercise 2

```r
directpoly <- function(x, coeff){
  n <- length(coeff)
  y <- 0
  for(i in 1:n) {
    y <- y + coeff[i]*x^(n-i)
  }
  return(y)
}
```

```r
directpoly(2,1:3)
```

```
## [1] 11
```

Exercise 3

```r
hornerpoly <- function(x, coeff){
  n <- length(coeff)
  a <- rep(coeff[1], length(x))
  for (i in (n-1):1) {
    a <- a*x + coeff[n-i+1]
  }
  return(a)
}
```

```r
hornerpoly(2,1:3)
```

```
## [1] 11
```