Solutions for ST340 Lab 0

2020–21

R Programming Exercises

These math / programming problems are the first 7 Project Euler problems. There are many more at http://projecteuler.net/

Problem 1: Multiples of 3 and 5

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23. Find the sum of all the multiples of 3 or 5 below 1000.

```
total.sum <- 0
for (i in 1:999) {
  if ((i %% 3 == 0) || (i %% 5 == 0)) {
    total.sum <- total.sum + i
  }
}
print(total.sum)
```

```
## [1] 233168
```

Problem 2: Even Fibonacci numbers

Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, … By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

```
fib.upto <- function(n) {
  N <- 1
  seq <- c(1,1)
  l <- 2
  while (N <= n) {
    N <- seq[l-1] + seq[l]
    seq <- c(seq,N)
    l <- l + 1
  }
  return(seq)
}

seq <- fib.upto(4e6)
s <- 0
for (i in 1:length(seq)) {
  if (seq[i] %% 2 == 0) {
    s <- s + seq[i]
  }
```

```
}
print(s)
```

```
## [1] 4613732
```

Problem 3: Largest prime factor

The prime factors of 13195 are 5, 7, 13 and 29. What is the largest prime factor of the number 600851475143?

```
find.factors <- function(n) {
  factors <- NULL
  factored <- FALSE
  p <- 2
  while (!factored) {
    if (n %% p == 0) {
      n <- n/p
      factors <- c(factors,p)
    } else {
      p <- p + 1
    }
    if (p > sqrt(n)) {
      factors <- c(factors,n)
      factored <- TRUE
    }
  }
  return(factors)
}
print(max(find.factors(600851475143)))
```

```
## [1] 6857
```

Problem 4: Largest palindrome product

A palindromic number reads the same both ways. The largest palindrome made from the product of two 2-digit numbers is $9009 = 91 \times 99$. Find the largest palindrome made from the product of two 3-digit numbers.

Hint: try `?strplit` to find an analogue of `rev()` for strings

```
strReverse <- function(x) sapply(lapply(strsplit(x, NULL), rev), paste, collapse = "")

is.palindrome <- function(n) {
  return(toString(n)==strReverse(toString(n)))
}

hits <- NULL
for (i in 900:999) {
  for (j in i:999) {
    n <- i*j
    if (is.palindrome(n)) {
      hits <- c(hits,n)
    }
  }
}
```

```r
print(max(hits))
```

```
## [1] 906609
```

```r
# Alternatively
found <- FALSE
for(i in seq(999,900,by=-1))
  for(j in seq(i,900,by=-1)) {
    n <- i*j
    if(is.palindrome(n)) {
      print(n)
      found <- TRUE
    }
    if(found) break
  }
```

```
## [1] 906609
```

Problem 5: Smallest multiple

2520 is the smallest number that can be divided by each of the numbers from 1 to 10 without any remainder. What is the smallest positive number that is evenly divisible by all of the numbers from 1 to 20?

```r
l_f <- lapply(3:20,find.factors)
factors <- numeric()
for (f in l_f) {
  p_f <- pmatch(f, factors)
  if (anyNA(p_f)) {
    miss_f <- which(is.na(p_f))
    for (i in miss_f) {
      factors <- sort(c(factors, f[i]))
    }
  }
}
n <- prod(factors)
print(n)
```

```
## [1] 232792560
```

```r
# alternatively
pf <- unlist(lapply(l_f,table))
prod.f <- 1
for(f in unique(names(pf))) {
  prod.f <- prod.f*as.numeric(f)^max(pf[f==names(pf)])
}
print(prod.f)
```

```
## [1] 232792560
```

Problem 6: Sum square difference

The sum of the squares of the first ten natural numbers is $1^2 + 2^2 + \cdots + 10^2 = 385$. The square of the sum of the first ten natural numbers is $(1 + 2 + \cdots + 10)^2 = 55^2 = 3025$. Hence the difference between the sum of the squares of the first ten natural numbers and the square of the sum is $3025 - 385 = 2640$.

Find the difference between the sum of the squares of the first one hundred natural numbers and the square of the sum.

```r
print(sum(1:100)^2 - sum((1:100)^2))
```

```
## [1] 25164150
```

Problem 7: 10001st prime

By listing the first six prime numbers: 2, 3, 5, 7, 11, and 13, we can see that the 6th prime is 13. What is the 10001st prime number?

```r
indivisible <- function(n,ms) {
  for (m in ms) {
    if (m > sqrt(n)) return(TRUE)
    if (n %% m == 0) return(FALSE)
  }
  return(TRUE)
}

next.prime <- function(primes) {
  n <- primes[length(primes)]
  while (TRUE) {
    n <- n + 1
    if (indivisible(n,primes)) {
      return(c(primes,n))
    }
  }
}

primes <- 2
for (i in 2:10001) {
  primes <- next.prime(primes)
}

print(primes[10001])
```

```
## [1] 104743
```