ESS212-HW1

Yiting Yu

2024-01-15

Yiting's github repository url

Problem 2

```
language <- "R"
cat(sprintf("%s says: Hello, World!\n", language))
```

```
## R says: Hello, World!
```

Problem 3

$$S_1(n) \equiv 1 + 2 + 3 + ... + n = \frac{n(n+1)}{2}$$
$$O_1(n) \equiv 1 + 3 + 5 + ... + (2n-1) = n^2$$
$$S_2(n) \equiv 1^2 + 2^2 + 3^2 + ... + n^2 = \frac{n(n+1)(n+2)}{6}$$

```
# S_1(n)
# iterative algorithm
S1_iter <- function(n) {
  S <- 0
  for (i in  1:n) {
    S <- S + i
    }
  return(S)
  }

# recursive algorithm
S1_rec <- function(n) {
  if (n > 1) {
    S = n + S1_rec(n-1)
    }
  else{
    S = 1
    }
  return(S)
  }
```

Verify

```r
n <- c(1, 10, 100)
(r_s1 <- data.frame(
  n = n,
  iter_result = sapply(n, S1_iter),
  rec_result = sapply(n, S1_rec),
  real_result = n*(n+1)/2
))
```

```
##     n iter_result rec_result real_result
## 1   1           1          1           1
## 2  10          55         55          55
## 3 100        5050       5050        5050
```

- Describe the internal state of the programs during the computational process.
- Using either the recursive algorithm or the iterative algorithm, write and test computer programs to evaluate O1(n) and S2(n).

```r
# O_1(n)
# iterative algorithm
O1_iter <- function(n) {
  S <- 0
  for (i in 1:n) {
    S <- S + 2*i-1
  }
  return(S)
}

# recursive algorithm
O1_rec <- function(n) {
  if (n > 1) {
    S = 2*n-1 + O1_rec(n-1)
  } else {
    S = 1
  }
}

# verify
n <- c(1, 10, 100)
(r_o1 <- data.frame(
  n = n,
  iter_result = sapply(n, O1_iter),
  rec_result = sapply(n, O1_rec),
  real_result = n^2
))
```

```
##     n iter_result rec_result real_result
## 1   1           1          1           1
## 2  10         100        100         100
## 3 100       10000      10000       10000
```

```r
# S_2(n)
## iterative algorithm
S2_iter <- function(n) {
  S <- 0
```

```r
  for (i in 1:n) {
    S <- S + i^2
  }
  return(S)
}
## recursive algorithm
S2_rec <- function(n) {
  if (n > 1) {
    S = n^2 + S2_rec(n-1)
    }
  else{
    S = 1
    }
  return(S)
  }

## verify
n <- c(1, 10, 100)
(r_s2 <- data.frame(
  n = n,
  iter_result = sapply(n, S2_iter),
  rec_result = sapply(n, S2_rec),
  real_result = (n*(n+1)*(2*n+1))/6
))
```

```
##     n iter_result rec_result real_result
## 1   1           1          1           1
## 2  10         385        385         385
## 3 100      338350     338350      338350
```

Problem 4

$$C(n,k) = C(n-1,k-1) + C(n-1,k)$$

- Write a procedure in your favorite computer language that uses Equation (7) evaluates $C(n,k)$ using a recursive algorithm.
- Write a procedure in your favorite computer language that takes an integer n as input and prints out the n-th row of Pascal's triangle.