Experiment 5: Recursions

## Recursions

### nCr

fact <- function(x) {  
 if(x <= 1) return(x) else return(x \* fact(x - 1))  
}  
nCr <- function(n, r) {  
 x <- fact(n) / (fact(n - r) \* fact(r))  
 return(x)  
}  
  
print(paste('15 items can be arranged in', nCr(15, 10), 'ways, when taking 10 items at a time'))

## [1] "15 items can be arranged in 3003 ways, when taking 10 items at a time"

### Quick Sort

swap <- function(vec, i, j) {  
 temp <- vec[i]  
 vec[i] <- vec[j]  
 vec[j] <- temp  
 return(vec)  
}  
partition <- function(vec, l, h) {  
 pivot <- vec[h]  
 i <- 1  
 j <- h - 1  
 while(vec[i] > pivot) {  
 if(vec[j] < pivot) {   
 vec <- swap(vec, i, j)  
 }  
 i <- i + 1  
 j <- j - 1  
 }  
 return(list(vec, i))  
}  
quicksort <- function(vec, l, h) {  
 results <- partition(vec, l, h)  
 vec <- results[1]  
 pivot <- results[2]  
 vec <- quicksort(vec, l, pivot - 1)  
 vec <- quicksort(vec, pivot + 1, h)  
 return(vec)  
}

### Binary Search

bsearch <- function(haystack, needle, lb, ub) {  
 mid <- (lb + ub) %/% 2  
 if(haystack[mid] == needle) {  
 return(needle)  
 } else if(lb == ub) {  
 return(-1) # not found  
 } else if(haystack[mid] > needle) {  
 return(bsearch(haystack, needle, mid + 1, ub))  
 } else if(haystack[mid] < needle) {  
 return(bsearch(haystack, needle, lb, mid - 1))  
 }  
}  
vec <- seq(1, 100, by = 13)  
vec

## [1] 1 14 27 40 53 66 79 92

index1 <- bsearch(vec, 10, 1, length(vec))  
print(paste('10 found at', index1))

## [1] "10 found at -1"

index2 <- bsearch(vec, 27, 1, length(vec))  
print(paste('27 found at', index2))

## [1] "27 found at -1"

## Row Wise and Column Wise Sum of Matrix

# find row sums  
rs <- function(mat1) {  
 n\_rows <- dim(mat1)[1]  
 for(i in seq(1,n\_rows)) {  
 print(paste('Sum of row is', sum(mat1[i,])))  
 }  
}  
# find column sums  
cs <- function(mat1) {  
 n\_cols <- dim(mat1)[2]  
 for(i in seq(1,n\_cols)) {  
 print(paste('Sum of column is', sum(mat1[,i])))  
 }  
}  
matX <- matrix(1:100, nrow = 20)  
print('Matrix is')

## [1] "Matrix is"

print(matX)

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1 21 41 61 81  
## [2,] 2 22 42 62 82  
## [3,] 3 23 43 63 83  
## [4,] 4 24 44 64 84  
## [5,] 5 25 45 65 85  
## [6,] 6 26 46 66 86  
## [7,] 7 27 47 67 87  
## [8,] 8 28 48 68 88  
## [9,] 9 29 49 69 89  
## [10,] 10 30 50 70 90  
## [11,] 11 31 51 71 91  
## [12,] 12 32 52 72 92  
## [13,] 13 33 53 73 93  
## [14,] 14 34 54 74 94  
## [15,] 15 35 55 75 95  
## [16,] 16 36 56 76 96  
## [17,] 17 37 57 77 97  
## [18,] 18 38 58 78 98  
## [19,] 19 39 59 79 99  
## [20,] 20 40 60 80 100

print('')

## [1] ""

rs(matX)

## [1] "Sum of row is 205"  
## [1] "Sum of row is 210"  
## [1] "Sum of row is 215"  
## [1] "Sum of row is 220"  
## [1] "Sum of row is 225"  
## [1] "Sum of row is 230"  
## [1] "Sum of row is 235"  
## [1] "Sum of row is 240"  
## [1] "Sum of row is 245"  
## [1] "Sum of row is 250"  
## [1] "Sum of row is 255"  
## [1] "Sum of row is 260"  
## [1] "Sum of row is 265"  
## [1] "Sum of row is 270"  
## [1] "Sum of row is 275"  
## [1] "Sum of row is 280"  
## [1] "Sum of row is 285"  
## [1] "Sum of row is 290"  
## [1] "Sum of row is 295"  
## [1] "Sum of row is 300"

print('')

## [1] ""

cs(matX)

## [1] "Sum of column is 210"  
## [1] "Sum of column is 610"  
## [1] "Sum of column is 1010"  
## [1] "Sum of column is 1410"  
## [1] "Sum of column is 1810"