

MSMS - 106

Ananda Biswas

Practical 09



Question : Consider a vector with values 3, 2, 5, 4, 4, 5, 7, 6. Implement

(i) Bubble sort,

(ii) Insertion sort,

(iii) Quick sort

in R.

```
x <- c(3, 2, 5, 4, 4, 5, 7, 6)
```

⊕ Bubble Sort

```
bubble_sort <- function(x){  
  i <- 1  
  while(i <= length(x)){  
    j <- i  
    while(j <= length(x)-1){  
      if(x[j] > x[j+1]){  
        x[c(j, j+1)] <- x[c(j+1, j)]  
      }  
      j <- j + 1  
    }  
    i <- i + 1  
  }  
  return(x)  
}
```

```
sorted_x <- bubble_sort(x)  
sorted_x
```

```
## [1] 2 3 4 4 5 5 6 7
```

⊕ Insertion Sort

```
insertion_sort <- function(x){
  i <- 1
  while(i <= length(x)){
    key <- x[i]
    j <- i - 1
    while(j >= 1){
      if(key < x[j]){
        x[c(i, j)] <- x[c(j, i)]
      }
      j <- j - 1
    }
    i <- i + 1
  }
  return(x)
}
```

```
sorted_x <- insertion_sort(x)
sorted_x

## [1] 2 3 4 4 5 5 6 7
```

⊕ Quick Sort

```
partition <- function(values, low, high){

  pivot <- values[low]

  i <- low; j <- high + 1

  while(i < j){
    repeat{
      i <- i + 1
      if(values[i] > pivot || i == high) break
    }

    repeat{
      j <- j - 1
      if(values[j] <= pivot) break
    }

    if(i < j) values[c(i, j)] <- values[c(j, i)]
  }

  values[c(low, j)] <- values[c(j, low)]

  return(list(j, values))
}
```

```
quick_sort <- function(values, low, high){  
  
  if(low >= high) return(values)  
  
  p <- partition(values, low, high)  
  partition_index <- p[[1]]  
  values <- p[[2]]  
  
  values <- quick_sort(values, low, partition_index - 1)  
  values <- quick_sort(values, partition_index + 1, high)  
  
}
```

```
sorted_x <- quick_sort(x, 1, length(x))  
sorted_x  
  
## [1] 2 3 4 4 5 5 6 7
```