Sorting Algorithms
++++++++++++++++++
1. Quicksort
2. Selection Sort
3. Buble Sort
4. Insertion Sort

```
ALGORITHM :: Insertion_Sort(a[0....n-1])
//Sorts the data in ascending order, using insertion sort
//INPUT  :An Array a[0.....n-1] of orderable elements
//OUTPUT :An Array a[0.....n-1] of ordered   elements

 for  i<-1 to n-1 do
      j<-i-1
      item <-a[i]

      while j>=0 && a[j]> item
            a[j+1]<-a[j]
            j--
      a[j+1]<-item
```

Program using java for insertion sort
+++++++++++++++++++++++++++++++++++++

```
import java.util.*;

class InsertionSortApp
{
      public static void main(String[] args)
      {
            Scanner scan = new Scanner(System.in);

            System.out.print("Enter the size of an array :: ");
            int n = scan.nextInt();

            //Creating an Array
            int[] arr = new int[n];

            //loop to read the array elements
            for (int i=0;i<=n-1;i++ )
            {
                  System.out.print("Enter the array element:: ");
                  arr[i] = scan.nextInt();
            }

            System.out.println("Array before Sorting :: "+Arrays.toString(arr));

            insertion_sort(arr,n);

            System.out.println("Array after Sorting :: "+Arrays.toString(arr));

      }

      public static void insertion_sort(int[] arr,int n)
      {
            for (int i=1;i<=n-1;i++ )
            {
                  int item = arr[i];
                  int j = i-1;

                  while (j>=0 && arr[j]>item)
                  {
                        arr[j+1] = arr[j];
                        j--;
                  }
                  arr[j+1] = item;
            }
      }
}
```

```
Output
Enter the size of an array :: 5
Enter the array element:: 105
Enter the array element:: 42
Enter the array element:: 12
Enter the array element:: 26
Enter the array element:: 87
Array before Sorting :: [105, 42, 12, 26, 87]
Array after  Sorting :: [12, 26, 42, 87, 105]
```

MergeSort
+++++++++

```
ALGORITHM :: Merge_Sort(a[0....n-1],0,n-1)
//Sorts the data in ascending order, using merge sort
//INPUT  :An Array a[0.....n-1] of orderable elements
//OUTPUT :An Array a[0.....n-1] of ordered   elements

   i <- low
   j <- mid+1
   k <- low

   while  i<=mid && j<=high do
      if a[i] < a[j]
           c[k] <- a[i]
            i++
            k++
      otherwise
           c[k] <- a[j]
            k++
            j++
    if a[i]>mid
      while j<=high do
            c[k] <- a[j]
            k++
            j++
    otherwise
      while i<=mid do
            c[k] <- a[i]
            i++
            k++
```

Program using Java
++++++++++++++++++

```
import java.util.*;

class MergeSortApp
```

```
{
      public static void main(String[] args)
      {
            Scanner scan = new Scanner(System.in);

            System.out.print("Enter the size of an array :: ");
            int n = scan.nextInt();

            //Created an Array
            int[] arr = new int[n];

            //loop to read the values into array
            for (int i=0;i<=n-1;i++ )
            {
                  System.out.print("Enter the array element:: ");
                  arr[i] = scan.nextInt();
            }

            System.out.println("Array before Sorting :: "+Arrays.toString(arr));

            merge_sort(arr,0,n-1);

            System.out.println("Array After  Sorting :: "+Arrays.toString(arr));
      }
      public static void merge_sort(int[] arr,int low,int high)
      {
            if (low<high)
            {
                  //splitting of an array
                  int mid = (low+high)/2;

                  merge_sort(arr,low,mid);
                  merge_sort(arr,mid+1,high);

                  //logic to perform mergesort
                  combine(arr,low,mid,high);
            }
      }
      public static void combine(int arr[],int low,int mid,int high)
      {
            int[] c =new int[high+1];

            //logic of merging both the arrays[bags]
            int i = low;
            int j = mid+1;
            int k = low;

            //Check whether both bag contains element or not
            while (i<=mid && j<=high)
            {
                  if (arr[i]< arr[j])
                  {
                        c[k]= arr[i];
                        i++;
                  }
                  else
                  {
                        c[k] = arr[j];
                        j++;
                        }
                        k++;
                }

                //1st bag empty
                if (i>mid)
                {
                        //copy all 2nd bag elements to new bag
                        while (j<=high)
                        {
                                c[k]=arr[j];
                                j++;
                                k++;
                        }
                }

                else
                {
                        //copy all 1st bag elements to new bag
                        while (i<=mid)
                        {
                                c[k]=arr[i];
                                i++;
                                k++;
                        }
                }

                //Copy from old bag to new bag
                for (k=low;k<=high;k++ )
                {
                        arr[k] = c[k];
                }
        }
}
```

```
Output
Enter the size of an array :: 8
Enter the array element:: 55
Enter the array element:: 33
Enter the array element:: 11
Enter the array element:: 99
Enter the array element:: 88
Enter the array element:: 22
Enter the array element:: 44
Enter the array element:: 77
Array before Sorting :: [55, 33, 11, 99, 88, 22, 44, 77]
Array After  Sorting :: [11, 22, 33, 44, 55, 77, 88, 99]
```

Softwares required
++++++++++++++++++

1. JDK(already installed)

2. IDE   :: eclipse
      https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2023-12/R/eclipse-jee-2023-12-R-win32-x86_64.zip

3. Database :: MySQL[password :: root123]
      https://dev.mysql.com/downloads/installer/

```
        https://sqlyog.en.download.it/downloading


Next class Tuesday :: 7.30PM to 10.30PM
 topic: Generics, Enum and Annotations
```