MERGE SORT ALGORITHM
DAA CSE 3004

19BCE7578
TARUN

CODE

```
import java.util.Scanner;
 public class MergeSort
{
void merge(int arr[], int l, int m, int r)
{
int n1 = m - l + 1;
int n2 = r - m;


  int L[] = new int[n1];
  int R[] = new int[n2];


  for (int i = 0; i < n1; ++i)
  L[i] = arr[l + i];
  for (int j = 0; j < n2; ++j)
  R[j] = arr[m + 1 + j];
  int i = 0, j = 0;
  int k = l;
  while (i < n1 && j < n2) {
  if (L[i] <= R[j]) {
  arr[k] = L[i];
```

```
i++;
}
else {
arr[k] = R[j];
j++;
}
k++;
}
while (i < n1) {
arr[k] = L[i];
i++;
k++;
}
while (j < n2) {
arr[k] = R[j];
j++;
k++;
}
}
void sort(int arr[], int l, int r)
{
if (l < r) {
int m =l+ (r-l)/2;
sort(arr, l, m);
sort(arr, m + 1, r);
```

```
merge(arr, l, m, r);

}

}

static void printArray(int arr[])

{

int n = arr.length;

for (int i = 0; i < n; ++i)

System.out.print(arr[i] + " ");

System.out.println();

}


public static void main(String args[])

{

long start = System.nanoTime();

int arr[] = {75, 10, 58, 33, 50, 59, 64, 40, 97, 34,58,65};


System.out.println("Given Array");

printArray(arr);

MergeSort ob = new MergeSort();

ob.sort(arr, 0, arr.length - 1);

System.out.println("\nSorted array");

printArray(arr);

long end = System.nanoTime();

long time = (end - start)/1000000;

System.out.println("Running time in milli seconds: "+time);
```

}

}

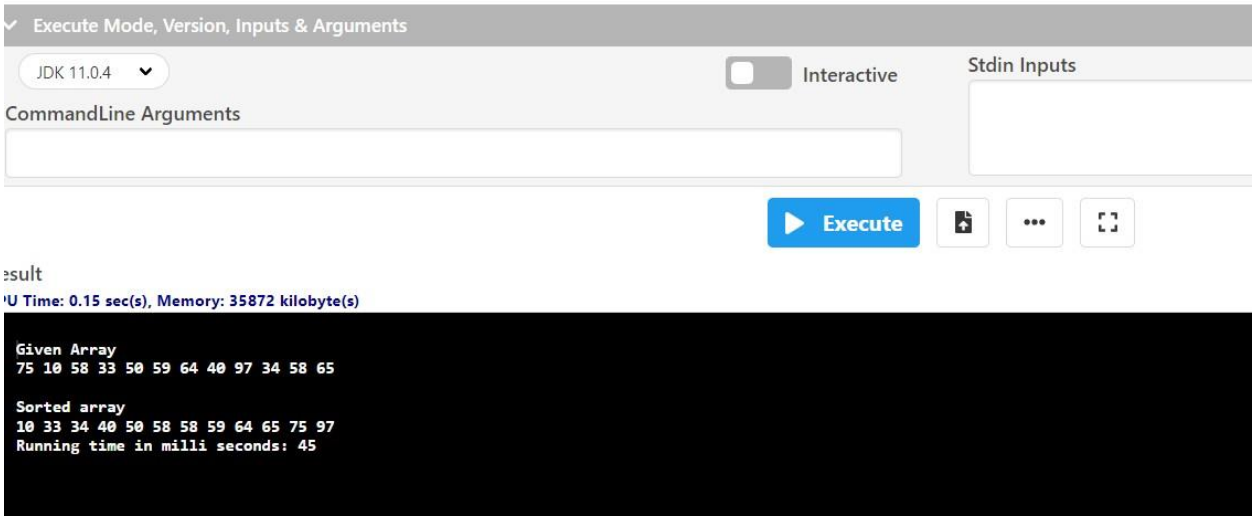
```
import java.util.Scanner;
 public class MergeSort
{
void merge(int arr[], int l, int m, int r)
{
int n1 = m - l + 1;
int n2 = r - m;

int L[] = new int[n1];
int R[] = new int[n2];

for (int i = 0; i < n1; ++i)
L[i] = arr[l + i];
for (int j = 0; j < n2; ++j)
R[j] = arr[m + 1 + j];
int i = 0, j = 0;
int k = l;
while (i < n1 && j < n2) {
if (L[i] <= R[j]) {
arr[k] = L[i];
i++;
}
else {
arr[k] = R[j];
j++;
}
k++;
}
while (i < n1) {
arr[k] = L[i];
i++;
k++;
}
while (j < n2) {
arr[k] = R[j];
j++;
k++;
}
}
void sort(int arr[], int l, int r)
{
if (l < r) {
int m =l+ (r-l)/2;
sort(arr, l, m);
sort(arr, m + 1, r);
merge(arr, l, m, r);
}
}
static void printArray(int arr[])
```

```
public static void main(String args[])
{
long start = System.nanoTime();
int arr[] = {75, 10, 58, 33, 50, 59, 64, 40, 97, 34,58,65};

System.out.println("Given Array");
printArray(arr);
MergeSort ob = new MergeSort();
ob.sort(arr, 0, arr.length - 1);
System.out.println("\nSorted array");
printArray(arr);
long end = System.nanoTime();
long time = (end - start)/1000000;
System.out.println("Running time in milli seconds: "+time);
}
}
```



Time complexity:

T(n)=total time

T(n)=2T(n/2)+n-1

T(n)=2T(n/2)+n ------1

n=n/2

T(n/2)=2T(n/4)+n/2---------2

Putting equ 2 in 1

T(n)=4T(n/4)+2n --------3

Putting n=n/4 in equ 1

T(n/4)=2T(n/4)+n/4----------4

Putting equ 4 in 3

T(n)= 8T(n/4)+3n----------5

Log n= i log 2

Logn/log2=i

=2*0+log2n.n

=T(n)

=nlogn