KIET Group of Institutions, Ghaziabad

Department of Computer Applications

(An ISO – 9001: 2015 Certified & 'A' Grade accredited Institution by NAAC)

Design and Analysis of Algorithm

RCA 352: Session 2020-21

DAA Lab

Experiment-No.3

Objective: Implement the MERGE-SORT algorithm to sort the given list of N numbers and plot graph.

Scheduled Date:	Compiled Date:	Submitted Date:
14-8-2020	24-8-2020	30-8-2020

Algorithm:

MERGE(A, p, q,r)

1. n1 ← q − p + 1

2. n2 ← r − q

3. create arrays L[1 . . n1 + 1] and R[1 . . n2 + 1]

4. for i ← 1 to n1

5. do L[i] ← A[p + i − 1]

6. for j ← 1 to n2

7. do R[j] ← A[q + j]

8. L[n1 + 1] ←

9. R[n2 + 1] ←

10. i ← 1

11. j ← 1

12. for k ← p to r

13. do if L[i] ≤ R[j]

14. then A[k] ← L[i]

15. i ← i + 1

16. else A[k] ← R[j]

17. j ← j + 1

MERGE-SORT(A, p,r)

Design and Analysis of Algorithm
RCA 352: Session 2020-21
DAA Lab

1. if p < r

2. then q ← (p + r)/2

3. MERGE-SORT(A, p, q)

4. MERGE-SORT(A, q + 1,r)

5. MERGE(A, p, q,r)

Program file insertion_sort.c :

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<alloc.h>
int count=0;
void merge(int[10],int,int,int);
void main()
{
        void getdata(int[10],int);
        void putdata(int[10],int);

        void merge_sort(int[10],int,int);
        int i,a[100],n;
        clrscr();
        printf("enter the value of n\n");
        scanf("%d",&n);
        getdata(a,n);
        printf("\nbefore soring\n");
```

```
        putdata(a,n);
        printf("hello");
        merge_sort(a,0,n-1);
        printf("hello");
        printf("\nafter sorting\n");
        putdata(a,n);
        printf("\n for n = %d value of count is  %d",n,count);
        getch();
}
void getdata(int a[10],int n)
{
    int k;
    printf("enter the value  for sorting\n");
    for(k=0;k<n;k++)
    {
     scanf("%d",&a[k]);
    }
}
void putdata(int a[10], int n)
{
        int k;
        for(k=0;k<n;k++)
        {
            printf("%d\t",a[k]);
         }
```

```
        printf("\n");
}
void merge_sort(int a[],int p,int r)
{
  int q;

        if(p<r)
         {

                q=(p+r)/2;

                merge_sort(a,p,q);
                merge_sort(a,q+1,r);
                merge(a,p,q,r);

         }
}
 void merge(int a[],int p, int q, int r)
{
  int n1,n2;
  int i,j,k;
 int l[100],r1[100];
   n1=q-p+1;
count++;
   n2=r-q;
```

```
count++;


 l[n1]=999;
 r1[n2]=999;


 for(i=0;i<n1;i++)
        {
count++;
          l[i]=a[p+i];
count++;
        }
count++;
  for(j=0;j<n2;j++)
        {
count++;
          r1[j]=a[q+j+1];
count++;
        }
count++;
   i=0;
count++;
   j=0;
count++;
  for(k=p;k<=r;k++)
    {
```

```
count++;
        if(l[i]<=r1[j])
        {
count++;
                a[k]=l[i];
count++;
                i=i+1;
count++;
        }
        else
        {
count++;
                a[k]=r1[j];
count++;
                j=j+1;
count++;
        }
count++;
   }
count++;


}
```

KIET Group of Institutions, Ghaziabad

Department of Computer Applications

(An ISO – 9001: 2015 Certified & 'A' Grade accredited Institution by NAAC)

Design and Analysis of Algorithm

RCA 352: Session 2020-21

DAA Lab

Output

Inputs	Best Case	Average Case	Worst Case
5	112	112	112
10	301	301	303
15	511	513	517
20	749	749	751
25	994	994	999

Graph