



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.2

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

Scheduled Date:	Compiled Date:	Submitted Date:
14-8-2020	18-8-2020	30-8-2020

Algorithm:

```
procedure quickSort(left, right)
```

```
1. if right-left less then 0
2.   return
3. else
4.   pivot = A[right]
5.   partition = partitionFunc(left, right, pivot)
6.   quickSort(left,partition-1)
7.   quickSort(partition+1,right)
8. end if
```

```
end procedure
```

```
function partitionFunc(left, right, pivot)
```

```
1. leftPointer = left
2. rightPointer = right - 1

3. while True do
4.   while A[++leftPointer] //pivot do
5.     //do-nothing
6.   end while

7.   while (rightPointer > 0 && A[--rightPointer] > pivot) do
8.     //do-nothing
9.   end while

10.  if (leftPointer >= rightPointer)
11.    break
12.  else
13.    swap (leftPointer,rightPointer)
14.  end if
```



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

```
15. end while
```

```
16. swap (leftPointer,right)
```

```
17. return (leftPointer)
```

```
end function
```

Program:

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<alloc.h>
int count=0;
int partition(int[10],int,int);
void main()
{
void getdata(int[10],int);
void putdata(int[10],int);
void quick_sort(int[10],int,int);
int i,a[100],n;
clrscr();
printf("Enter the Size of array=\n");
scanf("%d",&n);
getdata(a,n);
printf("\nBefore soring=\n");
putdata(a,n);
quick_sort(a,0,n-1);
printf("\nAfter sorting=\n");
putdata(a,n);
printf("\n For n = %d\n value of count is %d",n,count);
getch();
}
void getdata(int a[10],int n)
{
int k;
printf("Enter the %d Element for sorting\n",n);
for(k=0;k<n;k++)
{
scanf("%d",&a[k]);
}
}
```



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

```
void putdata(int a[10], int n)
{
    int k;
    for(k=0;k<n;k++)
    {
        printf("%d\t",a[k]);
    }
    printf("\n");
}

void quick_sort(int a[],int p,int r)
{
    int q;
    if(p<r)
    {
        count++;
        q=partition(a,p,r);
        count++;
        quick_sort(a,p,q-1);
        count++;
        quick_sort(a,q+1,r);
        count++;
    }
}

int partition(int a[],int p, int r)
{
    int x,i,j,temp;
    x=a[r];
    i=p-1;
    count++;
    for(j=p;j<=r-1;j++)
    {
        count++;
        if(a[j]<x)
        {
            count++;
            i=i+1;
            count++;
            temp=a[i];
            count++;
            a[i]=a[j];
            count++;
            a[j]=temp;
        }
    }
}
```



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

```
}  
count++;  
}  
count++;  
temp=a[i+1];  
count++;  
a[i+1]=a[r];  
count++;  
a[r]=temp;  
count++;  
return(i+1);  
}
```

Output

Input Size	Best Case	Average Case	Worst Case
5	96	63	72
10	351	143	251
15	523	404	387
20	650	305	611
25	861	497	766



KIET Group of Institutions, Ghaziabad

Department of Computer Applications

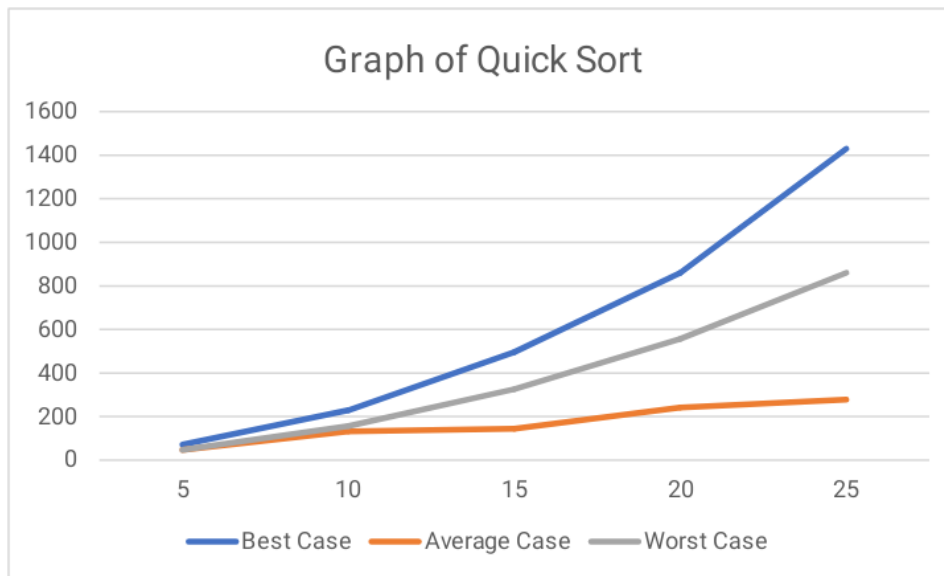
(An ISO – 9001: 2015 Certified & 'A' Grade accredited Institution by NAAC)

Design and Analysis of Algorithm

RCA 352: Session 2020-21

DAA Lab

Graph:



Conclusion

Case	Running Time : Growth of Function mathematically	Running Time : Growth of Function after observing graph
Best Case	$O(n \log n)$	$O(n \log n)$
Average Case	$O(n \log n)$	$O(n \log n)$
Worst Case	$O(n^2)$	$O(n^2)$