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.11

Objective: Implement the **ShellSort** algorithm to sort the given list of N numbers and plot graph

Scheduled Date:	Compiled Date:	Submitted Date:
5-9-2020	11-10-2020	12-10-2020

Algorithm:

Insertionsort(Input: Array A, Size N)

N: Number of values to be sort

A: Array of Size N

gap,i,j,temp: are variables

1. `for(gap=n/2;gap>=1;gap/=2)`
2. `for(i=gap;i<n;i++)`
3. `temp=A[i];`
4. `j=i-gap;`
5. `while(j>=0 && A[j]>temp)`
6. `A[j+gap]=A[j];`
7. `j=j-gap;`
8. `A[j+gap]=temp;`

Program to implement ShellSort

```
#include <stdio.h>
#include<stdlib.h>
void ShellSort(int [],int);
static int count=0;
int main()
{
    int a[30],n;
    printf("Enter the size of array that should be less than 30: ");
    scanf("%d",&n);
    printf("Enter the array elements\n");
    for(int i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    ShellSort(a,n);
    for(int i=0;i<n;i++)
    {
        printf("%d ",a[i]);
    }
    printf("\nFor n=%d no. of counts are=%d",n,count);
    return 0;
```

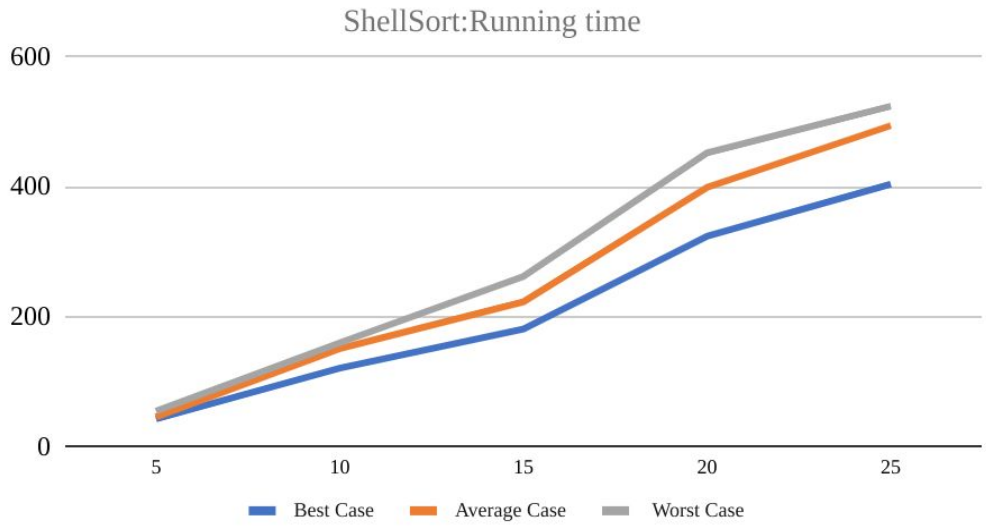
```
}
void swap(int *x,int *y)
{
 int temp=*x;
 *x=*y;
 *y=temp;
}
void ShellSort(int A[],int n)
{
    count++;
    int gap,i,j,temp;
     count++;
    for(gap=n/2;gap>=1;gap/=2)
    {
       count++;
        count++;
    for(i=gap;i<n;i++)
    {
          count++;
          count++;
        temp=A[i];
         count++;
        j=i-gap;
         count++;
 while(j>=0 && A[j]>temp)
    {
          count++;
        A[j+gap]=A[j];
         count++;
        j=j-gap;
         count++;

    }

 A[j+gap]=temp;
         count++;

    }
     count++;

 }

}
```

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

Output

Input Size	Best Case	Average Case	Worst Case
5	43	46	55
10	121	151	160
15	181	223	262
20	324	399	452
25	404	494	524

Graph



Conclusion

Case	Running Time : Growth of Function mathematically	Running Time : Growth of Function after observing graph
Best Case	O(n)	O(n)
Average Case	O(n3/2)	O(n3/2)

Worst Case	O($n3/2$)	O(n3/2)