# 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.10

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

| Scheduled Date: | Compiled Date: | Submitted Date: |
|---|---|---|
| 25-09-20 | 25-09-20 | 25-09-20 |

**Algorithm:**

**ImprovisedBubbleSort**( Input: Array A, Size N)

N: Number of values to be sort<br>
A: Array of Size N<br>
Flag,Temp, Pass,J : extra variable

```
Pass=1
while(pass<=n) do:
    J :=1;
    flag=0;
    while(j<=n-pass)  do:
        if(a[j]>a[j+1])
                temp :=a[j];
                a[j]    :=a[j+1];
                a[j+1]  :=temp;
                flag=1;
          j := j+1
        end while
          if(flag==0)
        break;
        pass := pass+1
end while
```

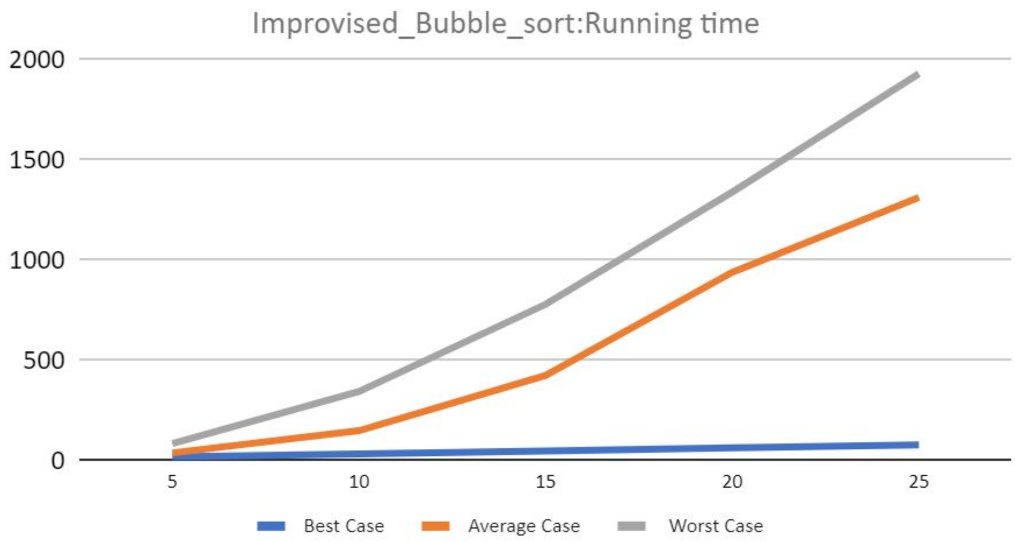
**Program file Improvised_bubble_sort.c :**

```
#include <stdio.h>
#include <stdlib.h>

int count = 0;
int main(){
    void get_data(int [],int);
    void bubble_sort(int[],int);
    void put_data(int[],int);
    int a[40];
    int n;
    printf("Enter the size of array should be less than 40:\n");
    scanf("%d",&n);
    get_data(a,n);
    printf("\nbefore sorting\n");
    put_data(a,n);
    bubble_sort(a,n);
    printf("\nafter sorting\n");
    put_data(a,n);
    printf("\n\n\nFor n=%d no.of counts=%d",n,count);

    return 0;
}

void get_data(int a[],int n)
{
    int i;
    printf("Enter the values of an array:\n");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }

}

void bubble_sort(int a[],int n)
{
       int pass,j,temp,flag;
       count++;

       for(pass=1;pass<=n-1;pass++)
       {
         count++;
         count++;
         flag=0;
         count++;
         for(j=0;j<n-pass;j++)
         {
          count++;
          count++;
          if(a[j]>a[j+1])
          {
           count++;
           temp=a[j];
           count++;
           a[j]=a[j+1];
           count++;
           a[j+1]=temp;
           flag=1;
           count++;
          }
          count++;
         }
         if(flag==0)
         {
             count++;
        break;
         }
        }
}


void put_data(int a[],int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%d\t",a[i]);
    }

}
```

**Output**

| Input Size | Best Case | Average Case | Worst Case |
|---|---|---|---|
| 5 | 17 | 37 | 83 |
| 10 | 32 | 148 | 343 |
| 15 | 47 | 423 | 778 |
| 20 | 62 | 938 | 1338 |
| 25 | 77 | 1311 | 1928 |

**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( $n^2$ ) | O( $n^2$ ) |
| Worst Case | O( $n^2$ ) | O( $n^2$ ) |