Bubble Sort:

```
package bubblesort;
public class Bubblesort {
  public static void main(String[] args) {
     Bubblesort obj=new Bubblesort();
     int arr[]={11,10,21,13,45,6,7,8,15,1};
     obj.bubblesort(arr);
     System.out.println("Sorted Array");
     obj.printarray(arr);
    }
  public  void bubblesort(int arr[]){
     int n=arr.length;
     for(int i=0;i<n-1;i++)
        for(int j=0;j<n-i-1;j++){
           if(arr[j]>arr[j+1]){
              int temp=arr[j];
              arr[j]=arr[j+1];
              arr[j+1]=temp;
           }
        }
     }
  void printarray(int arr[]){
     int n=arr.length;
     for(int i=0;i<n;i++){
        System.out.print(arr[i] +" ");
     }
  }
  }
```

Time complexity:

In bubble sort n-1 compression will be in first term, n-2 in second and n-3 and so on. So total number of compression is, (n-1) + (n-2) + (n-3) + …. + 3 + 2 + 1

Or, $\frac{n(n-1)}{2}$

Ignoring all constant, we can write time complexity O(n^2) and this is for worst and average case. But in best case it is O(n). Because array is already sorted.

Practice Problem

1. You have some of random data, then apply the bubble sort and print the data in descending order.

```
package descending;

import java.util.Scanner;

public class Descending {

public static void main(String[] args) {

      int i,j;

      Scanner obj= new Scanner(System.in);

      System.out.print("Enter Array Size:");

      int n=obj.nextInt();

     int arr[]=new int[n];

     System.out.println("Enter the Data:");

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

          arr[i]=obj.nextInt();

       }

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

          for(j=0;j<n-1;j++){

             if(arr[j]<arr[j+1]){

                int temp=arr[j];

                arr[j]=arr[j+1];

                arr[j+1]=temp;

             }
```

```
            }
        }
        System.out.print("Descending Order:");
        for(i=0;i<n;i++){
            System.out.print(arr[i]+" ");
        }
    }
```

2. Suppose, you have some of various data, then apply bubble sort and print the data ascending order and also print the sum of data.

```
package ascending;
import java.util.Scanner;
public class Ascending {
  public static void main(String[] args) {
        int i,j,sum=0;
        Scanner obj= new Scanner(System.in);
        System.out.print("Enter Array Size:");
        int n=obj.nextInt();
       int arr[]=new int[n];
       System.out.println("Enter the Data:");
         for(i=0;i<n;i++){
            arr[i]=obj.nextInt();
        }
        for(i=0;i<n;i++){
            for(j=0;j<n-1;j++){
                if(arr[j]>arr[j+1]){
                    int temp=arr[j];
                    arr[j]=arr[j+1];
                    arr[j+1]=temp;
                }
```

```
            }
        }
        System.out.print("Ascending Order:");
        for(i=0;i<n;i++){
            sum=sum+arr[i];
            System.out.print(arr[i]+" ");
        }
        System.out.println("");
        System.out.println("Sum="+sum);
       }
 }
```

Insertion Sort:

```
package insertationsort;
public class Insertationsort {
 // A funcation to sort array using insertation sort.
   void sort(int arr[]){
       int n=arr.length;
       for(int i=1;i<n;i++){
       int key=arr[i];
       int j=i-1;
       while(j>=0 && arr[j]>key){
           arr[j+1]=arr[j];
           j=j-1;
       }
       arr[j+1]=key;
       }
   }
   static void printArray(int arr[])
```

```
  {
     int n=arr.length;
     for(int i=0;i<n;i++)
        System.out.println(arr[i]+" ");
     System.out.println();
  }
 public static void main(String[] args)
  {
     int arr[]={12,11,13,5,6};
     Insertationsort obj=new Insertationsort();
     obj.sort(arr);
     printArray(arr);
  }
}
```

Time complexity:

```
     void sort(int arr[]){
     int n=arr.length;
     for(int i=1;i<n;i++){                    c1n
     int key=arr[i];                          c2(n-1)
     int j=i-1;                               c3(n-1)
     while(j>=0 && arr[j]>key){               c4n(n+1)/2
        arr[j+1]=arr[j];                      c5n(n+1)/2
        j=j-1;                                c6n(n+1)/2
     }
     arr[j+1]=key;                            c7(n-1)
     }
  }
```

Total time complexity: c1n+c2(n-1)+c3(n-1)+c4n(n+1)/2+c5(n+1)/2+c6n(n+1)/2+c7(n-1)

Best Case: In best case c4,c5,c6 is not execute.
c1n+c2n-c2+c3n-c3
=n(c1+c2) – (c2+c3)
This is look like y=an-b
Worst Case:
c1n+c2(n-1)+c3(n-1)+c4n(n+1)/2+c5(n+1)/2+c6n(n+1)/2+c7(n-1)
n^2(c4+c5+c6)/2+ n(c2+c3+c4+c5+c6)+(-c7)
Which look like: ax^2+bx+c

Best Case: O(n)
Worst Case: O(n^2)

Practice problem

1. You have some random data your job is that you have to sort them in descending order using Insertion Sort.

```
package descending;

import java.util.Scanner;

public class Descending {

public static void main(String[] args) {

    int i,j,sum=0;

   Scanner obj= new Scanner(System.in);

   System.out.print("Enter the Array:");

   System.out.println("");

   int n=obj.nextInt();

    int arr[]=new int[n];

  System.out.print("Enter the Data:");

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

      arr[i]=obj.nextInt();
```

```
        }
        int key;
        for(j=0;j<n;j++){
            key=arr[j];
            i=j-1;
            while(i>=0 && key>arr[i]){
                arr[i+1]=arr[i];
                i--;
            }
            arr[i+1]=key;
        }
        System.out.println("Ascending Order:");
        for(i=0;i<n;i++){

            sum=sum+arr[i];
            System.out.print(arr[i]+" ");
        }
    }
```

2.Suppose, you have some of various data, then apply insertion sort & print the data ascending order. Now, you have sorted data, Find the difference between maximum data and minimum data and also print the Difference of max and min.

```
package ascending;
import java.util.Scanner;
public class Ascending {
 public static void main(String[] args) {
        int i,j;
       Scanner obj= new Scanner(System.in);
       System.out.print("Enter the Array:");
```

```
        System.out.println("");
        int n=obj.nextInt();
       // n=obj.nextInt();
       int arr[]=new int[n];
       System.out.print("Enter the Data:");
        for(i=0;i<n;i++){
            arr[i]=obj.nextInt();
     }
        int key;
        for(j=0;j<n;j++){
            key=arr[j];
            i=j-1;
            while(i>=0 && key<arr[i]){
                arr[i+1]=arr[i];
                i--;
            }
            arr[i+1]=key;
        }
        System.out.println("Ascending Order:");
        for(i=0;i<n;i++){
            System.out.print(arr[i]+" ");
        }
    System.out.println("");
        int x=arr[n-1]-arr[0];
        System.out.println("Difference btn max-min="+x);
}
```

3. Suppose, you know two sorting algorithms one is Bubble sort & second is Insertion sort. Then applying the better algorithm in random data, print sorted data and also find the average of data.

```
package insertationsort;
import java.util.Scanner;
public class InsertationSort {
 public static void main(String[] args) {
          int i,j,sum=0;
        Scanner obj= new Scanner(System.in);
        System.out.print("Enter the Array:");
        System.out.println("");
        int n=obj.nextInt();
       // n=obj.nextInt();
       int arr[]=new int[n];
       System.out.print("Enter the Data:");
        for(i=0;i<n;i++){
            arr[i]=obj.nextInt();
           }
        int key;
        for(j=0;j<n;j++){
            key=arr[j];
            i=j-1;
            while(i>=0 && key<arr[i]){
                arr[i+1]=arr[i];
                i--;
            }
            arr[i+1]=key;
        }
        System.out.println("Ascending Order:");
        for(i=0;i<n;i++){
             sum=sum+arr[i];
```

```
        System.out.print(arr[i]+" ");
    }
    System.out.println("");
    int x=sum/n;
    System.out.println("Avg="+x);
}
```