

```
import java.lang.Math;
import java.util.HashMap;
import java.util.Map;

public class InsertionSort {

    int getRandomNumber(int min, int max) {
        // Get a random number
        return (int) (Math.random() * (max - min)) + min;
    }

    int[] getInitializedRandomArray(int[] arr, int size) {
        // Inserting random elements into the array
        for(int i=0; i<size; i++) {
            arr[i] = this.getRandomNumber(0, 500);
        }

        return arr;
    }

    void printArray(int[] arr, int size) {
        for(int j=0; j<size; j++) {
            System.out.print(arr[j]);

            // To not print ',' after the final element
            if (j!=size-1) {
                System.out.print(", ");
            }
        }
    }

    public static void main(String[] args) {
        // Initializing the insertion sortObj to use the class methods
        InsertionSort insertionSortObj = new InsertionSort();

        // Setting the min and max number of sets/arrays
        int minNumberOfSets = 10, maxNumberOfSets = 13;

        // Determining the total number of sets or arrays to be sorted
```

```

        int totalNumberOfSets =
insertionSortObj.getRandomNumber(minNumberOfSets, maxNumberOfSets);
        System.out.println("Total number of sets: " + totalNumberOfSets + "\n");

        // Initializing an empty array
        int[] arr = new int[60];

        // Setting the min and max number of elements for each array and
        declaring total number of elements
        int minNumberOfElements=30, maxNumberOfElements=60,
totalNumberOfElements;

        // Declaring the variables to be used for sorting
        int key, j;

        // Declaring the counter to maintain the actual count of instructions being
run
        int counter;

        // Initializing a map to maintain a record of the number of inputs and the
        number of instructions taken to sort the array
        Map<Integer, Integer> inputToActualCountMap = new HashMap<>();

        while(totalNumberOfSets > 0) {
            // Determining the total number of elements for an array using a
randomizer
            totalNumberOfElements =
insertionSortObj.getRandomNumber(minNumberOfElements,
maxNumberOfElements);
            System.out.println("Total number of elements: " +
totalNumberOfElements);

            // Get the initialized randomizer array
            arr = insertionSortObj.getInitializedRandomArray(arr,
totalNumberOfElements);

            System.out.print("Unsorted array: ");
            insertionSortObj.printArray(arr, totalNumberOfElements);

            // Sorting the array
            counter = 0;

```

```

    for(int i=1; i<totalNumberOfElements; i++) {
        // Incrementing the counter for the first time the 'i' counter is
        initialized and everytime it is incremented
        counter++;

        key = arr[i];
        // Incrementing the counter for everytime the 'key' variable is
        assigned
        counter++;

        j = i-1;
        // Incrementing the counter for everytime the 'j' counter is assigned
        counter++;

        while(j>=0 && arr[j] > key) {
            // Incrementing the counter everytime the comparison occurs
            counter++;

            arr[j+1] = arr[j];
            // Incrementing the counter everytime a number is moved/
            swapped to the right
            counter++;

            --j;
            // Incrementing the counter everytime the counter 'j' is
            decremented
            counter++;
        }
        // Incrementing the counter here because before exiting the loop the
        comparison will be carried out once
        counter++;

        arr[j+1] = key;
        // Incrementing the counter everytime 'key' is assigned to the array
        index 'j+1'
        counter++;
    }
    // Incrementing the counter here because before exiting the loop the
    comparison will be carried out once

```

```

        counter++;

        System.out.println();
        System.out.print("Sorted array: ");
        insertionSortObj.printArray(arr, totalNumberOfElements);

        inputToActualCountMap.put(totalNumberOfElements, counter);

        System.out.println("\n");
        totalNumberOfSets--;
    }

    System.out.println("\n");

    System.out.println(String.format("%10s %25s %10s %23s %10s", "N",
    "|", "Actual Count", "|", "Worst case T(N)"));
    System.out.println(String.format("%s",
    "-----"
    -----));

    inputToActualCountMap.forEach((input, count) -> {
        System.out.println(String.format("%10d %25s %10d %25s %10d",
input, "|", count, "|", (input * input)));
    });
}
}

```