Assignment 4

Sayak Ghorai || BT21GCS004 || B2 || Design and Analysis of Algorithm

Q: Write an algo for bubble sort for an array of integers.

Code:

BubbleSort.java

```java
import java.util.*;
public class BubbleSort {
    public static void main(String[] args) {
        int[] Arr1 = CreateArr();
        int[] Arr2 = CreateArr();
        int[] Arr3 = CreateArr();
        System.out.println("Original Array1: " + Arrays.toString(Arr1));
        System.out.println("Original Array2: " + Arrays.toString(Arr2));
        System.out.println("Original Array2: " + Arrays.toString(Arr2));
        int[][] Sorted1=myBubbleSort(Arr1);
        int[][] Sorted2=myBubbleSort(Arr2);
        int[][] Sorted3=myBubbleSort(Arr3);
        printRes(Sorted1,Sorted2,Sorted3);
    }
    3 usages
    static int[] CreateArr(){
        Scanner sc=new Scanner(System.in);
        System.out.println("Size of Arr?");
        int N=sc.nextInt();
        int[] Arr= new int[N];
        for(int i=0;i<N;i++){
            System.out.print("Enter "+(i+1)+"'th Element: ");
            Arr[i]=sc.nextInt();
        }
        return Arr;
    }
    3 usages
    static int[][] myBubbleSort(int[] Arr) {
        int SwapCount=0;
        for (int i = 0; i < Arr.length; i++) {
            for (int j = 0; j < Arr.length - i - 1; j++) {
                if (Arr[j] > Arr[j + 1]) {
                    int temp = Arr[j];
                    Arr[j] = Arr[j + 1];
                    Arr[j + 1] = temp;
                    SwapCount++;
                }
            }
        }
        return new int[][]{Arr,new int[] {SwapCount,Arr.length}};
    }
    1 usage
    1 usage
    static void printRes(int[][] Sorted1,int[][] Sorted2, int[][] Sorted3){
        int[] Arr1=Sorted1[0];
        int[] Arr2=Sorted2[0];
        int[] Arr3=Sorted3[0];
        //Calculate Best case
        if(Sorted1[1][0]==0){
            System.out.println("Best Case is Arr1, Swapped required : "+Sorted1[1][0]+" Sorted Array is: "+Arrays.toString(Arr1));
        }
        if (Sorted2[1][0]==0) {
            System.out.println("Best Case is Arr2, Swapped required : "+Sorted2[1][0]+" Sorted Array is: "+Arrays.toString(Arr2));
        }
        if (Sorted3[1][0]==0) {
            System.out.println("Best Case is Arr3, Swapped required : "+Sorted3[1][0]+" Sorted Array is: "+Arrays.toString(Arr3));
        }
        //Average Case
        if(Sorted1[1][0]<(((Sorted1[1][1])*(Sorted1[1][1]-1))/2)&&Sorted1[1][0]!=0){
            System.out.println("Average Case is Arr1, Swapped required : "+Sorted1[1][0]+" Sorted Array is: "+Arrays.toString(Arr1));
        }
        if (Sorted2[1][0]<(((Sorted2[1][1])*(Sorted2[1][1]-1))/2)&&Sorted2[1][0]!=0) {
            System.out.println("Average Case is Arr2, Swapped required : "+Sorted2[1][0]+" Sorted Array is: "+Arrays.toString(Arr2));
        }
        if (Sorted3[1][0]<(((Sorted3[1][1])*(Sorted3[1][1]-1))/2)&&Sorted3[1][0]!=0) {
            System.out.println("Average Case is Arr3, Swapped required : "+Sorted3[1][0]+" Sorted Array is: "+Arrays.toString(Arr3));
        }
        //Worst case
        if(Sorted1[1][0]==(((Sorted1[1][1])*(Sorted1[1][1]-1))/2)){
            System.out.println("Worst Case is Arr1, Swapped required : "+Sorted1[1][0]+" Sorted Array is: "+Arrays.toString(Arr1));
        }
        if (Sorted2[1][0]==(((Sorted2[1][1])*(Sorted2[1][1]-1))/2)) {
            System.out.println("Worst Case is Arr2, Swapped required : "+Sorted2[1][0]+" Sorted Array is: "+Arrays.toString(Arr2));
        }
        if (Sorted3[1][0]==(((Sorted3[1][1])*(Sorted3[1][1]-1))/2)) {
            System.out.println("Worst Case is Arr3, Swapped required : "+Sorted3[1][0]+" Sorted Array is: "+Arrays.toString(Arr3));
        }
    }
}
```

Output:

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java -Dvisualgc.id=215898072048500 -javaagent:/Applica
 .jar=63145:/Applications/IntelliJ IDEA CE.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.s
 /Users/sayakghorai/Desktop/DAA_Assignments/Daa_Assignment4/out/production/Daa_Assignment4 BubbleSort
Size of Arr?
8
Enter 1'th Element: 5
Enter 2'th Element: 4
Enter 3'th Element: 3
Enter 4'th Element: 6
Enter 5'th Element: 1
Enter 6'th Element: 2
Enter 7'th Element: 7
Enter 8'th Element: 8
Size of Arr?
9
Enter 1'th Element: 9
Enter 2'th Element: 8
Enter 3'th Element: 7
Enter 4'th Element: 6
Enter 5'th Element: 5
Enter 6'th Element: 4
Enter 7'th Element: 3
Enter 8'th Element: 2
Enter 9'th Element: 1
Size of Arr?
6
Enter 1'th Element: 1
Enter 2'th Element: 2
Enter 3'th Element: 3
Enter 4'th Element: 4
Enter 5'th Element: 5
Enter 6'th Element: 6
Original Array1: [5, 4, 3, 6, 1, 2, 7, 8]
Original Array2: [9, 8, 7, 6, 5, 4, 3, 2, 1]
Original Array2: [9, 8, 7, 6, 5, 4, 3, 2, 1]
Best Case is Arr3, Swapped required : 0 Sorted Array is: [1, 2, 3, 4, 5, 6]
Average Case is Arr1, Swapped required : 11 Sorted Array is: [1, 2, 3, 4, 5, 6, 7, 8]
Worst Case is Arr2, Swapped required : 36 Sorted Array is: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Process finished with exit code 0
```