“I have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of **0** for the involved assignment for my first offense and that I will receive a grade of **“F” for the course** for any additional offense.”
Name: Chinmay Kelkar

Instruction Counts:

Insertion Sort:

```
for (i = 1; i < size; i++){
                    for (j = i; j >= 0 && input[j] < input[j-1]←(barometric operation); j--){
                              temp= input[j];
                              input[j]   = input[j-1];
                              input[j-1] = temp;
                       }
}
```

$$\sum_{i=1}^{n}\sum_{j=i}^{0} 1 = \sum_{i=1}^{n} i = \frac{n(n-1)}{2} = \frac{n^2-1}{2} \quad \textit{which is } \theta(n^2)$$

Counting Sort:

```
for(i=0;i<size;i++)

countarray[input[i]]=countarray[input[i]] + 1; ←(Barometric Operation)
```

Let size=n

$$\sum_{i=0}^{n} 1 = n - - - - - - - - (1)$$

```
for(i=1;i<=max;i++)

countarray[i]=countarray[i]+countarray[i-1];←(Barometric Operation)
```

let max=n

$$\sum_{i=0}^{n} 1 = n - - - - - - - - (2)$$

```
for(i=0;i<size;i++){

        resultarray[countarray[input[i]]]=input[i];

        countarray[input[i]]=countarray[input[i]]-1;←(Barometric Operation)

                if(flag==1){

                        printf("\nFilling result array according to countarray");

                        animation(resultarray,0,size);

                }

}
```

Let size=n

$$\sum_{i=0}^{n} 1 = n - - - - - - - - (3)$$

Adding (1) (2) (3) we get addition as 3n which is $\theta(n)$

Merge Sort

```
if(low<high){ ←(Barometer operation)
            mid=(low+high)/2;
            partition(arr,low,mid);
             partition(arr,mid+1,high);
              merge(arr,low,mid,high);
}
```

Above barometric operation operates log n times --------(1)

```
while((l<=mid)&&(m<=high)){←(barometric operation)
  if(arr[l]<=arr[m]){
            temparray[i]=arr[l];
            l++;
        }
        else{
            temparray[i]=arr[m];
            m++;
        }
        i++;
    }
```

Above while loop will operate high times. Let high is n

$$\sum_{i=0}^{n} 1 = n - - - - - - - - (2)$$

From (1) and (2) we get $\theta(nlogn)$