

## Assignment 10

Name :- Varzil Thakkar

Roll No :- 21BCP090

### Implementing the Bubble Sort

```
def bubbleSort(arr):  
    n = len(arr)  
  
    # For loop to traverse through all element in an array  
    for i in range(n):  
        for j in range(0, n - i - 1):  
  
            # Range of the array is from 0 to n-i-1  
            # Swap the elements if the element found is greater than  
the adjacent element  
            if arr[j] > arr[j + 1]:  
                arr[j], arr[j + 1] = arr[j + 1], arr[j]  
  
def printArr(arr):  
    for i in range(len(arr)):  
        print("%d" % arr[i])  
  
myArr=[9,8,7,6,5,4,3,2,1]  
bubbleSort(myArr)  
printArr(myArr)
```

1  
2  
3  
4  
5  
6  
7  
8  
9

### Implementing the Selection Sort

```
def selectionSort(array):  
    size=len(array)  
    for s in range(size):  
        min_idx = s  
  
        for i in range(s + 1, size):  
  
            # For sorting in descending order for minimum element in
```

*each loop*

```
    if array[i] < array[min_idx]:  
        min_idx = i
```

*# Arranging min at the correct position*

```
(array[s], array[min_idx]) = (array[min_idx], array[s])
```

```
myArr=[9,8,7,6,5,4,3,2,1]  
selectionSort(myArr)
```

```
printArr(myArr)
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9
```

**Implementing the Insertion Sort**

```
def insertion_sort(list1):
```

*# Outer loop to traverse on len(list1)*

```
    for i in range(1, len(list1)):
```

```
        a = list1[i]
```

*# Move elements of list1[0 to i-1], which are greater to  
one position*

*# ahead of their current position*

```
    j = i - 1
```

```
    while j >= 0 and a < list1[j]:
```

```
        list1[j + 1] = list1[j]
```

```
        j -= 1
```

```
    list1[j + 1] = a
```

```
myArr=[9,8,7,6,5,4,3,2,1]  
insertion_sort(myArr)  
printArr(myArr)
```

```
1  
2  
3  
4  
5
```

6  
7  
8  
9

### Implementing the Merge Sort

```
def merge(arr, l, m, r):  
    n1 = m - l + 1  
    n2 = r - m  
  
    # create temp arrays  
    L = [0] * (n1)  
    R = [0] * (n2)  
  
    # Copy data to temp arrays L[] and R[]  
    for i in range(0, n1):  
        L[i] = arr[l + i]  
  
    for j in range(0, n2):  
        R[j] = arr[m + 1 + j]  
  
    # Merge the temp arrays back into arr[l..r]  
    i = 0      # Initial index of first subarray  
    j = 0      # Initial index of second subarray  
    k = l      # Initial index of merged subarray  
  
    while i < n1 and j < n2:  
        if L[i] <= R[j]:  
            arr[k] = L[i]  
            i += 1  
        else:  
            arr[k] = R[j]  
            j += 1  
        k += 1  
  
    # Copy the remaining elements of L[], if there  
    # are any  
    while i < n1:  
        arr[k] = L[i]  
        i += 1  
        k += 1  
  
    # Copy the remaining elements of R[], if there  
    # are any  
    while j < n2:  
        arr[k] = R[j]  
        j += 1  
        k += 1  
  
    # l is for left index and r is right index of the
```

*# sub-array of arr to be sorted*

```
def mergeSort(arr, l, r):  
    if l < r:  
        # Same as (l+r)//2, but avoids overflow for  
        # large l and h  
        m = l+(r-l)//2  
  
        # Sort first and second halves  
        mergeSort(arr, l, m)  
        mergeSort(arr, m+1, r)  
        merge(arr, l, m, r)
```

```
myArr=[9,8,7,6,5,4,3,2,1]  
mergeSort(myArr,0,len(myArr)-1)  
printArr(myArr)
```

1  
2  
3  
4  
5  
6  
7  
8  
9

```
# merge sort used to sort some values in between EXAMPLE:  
myArr=[9,8,7,6,5,4,3,2,1]  
mergeSort(myArr,3,len(myArr)-4)  
printArr(myArr)  
#this will sort the elements from position 4(3+1) to 6(10-4)
```

9  
8  
7  
4  
5  
6  
3  
2  
1