G. H. RAISONI COLLEGE OF ENGG., NAGPUR
(An Autonomous Institute under UGC Act 1956)
Department of Computer Science & Engg.

Date: 07/07/2020

Practical Subject: Design and Analysis of Algorithms
Session: 2020-21

Student Details:

Roll Number	49
Name	Shivam Tawari
Semester	3
Section	A
Branch	Artificial Intelligence

Practical Details: Practical Number- 02

Practical Aim	To Implement and analyze time complexity of Sorting Algorithm.
Theory & Algorithm	**Selection Sort:** ***Theory:*** Selection Sort is about selecting the smallest element from the list and placing it in the sorted portion of the list. Initially, the first element is considered the minimum and compared with other elements. During these comparisons, if a smaller element is found then that is considered the new minimum. After completion of one full round, the smallest element found is swapped with the first element. This process continues till all the elements are sorted. ***Algorithm:*** Step 1: START Step 2: Set i = min_value and rest elements unsorted. Step 3: If j < i, then set j = min_value and repeat the Step 3.

	Step 4: After one iteration, swap the min_value at first element position. Step 5: Now consider second element as i and repeat the steps until all the elements are covered. Step 6: STOP **Bubble Sort:** ***Theory:*** Bubble Sort is one of the most basic and simple algorithms. It takes in an unsorted list and keeps comparing each element with its right-side element in order to sort the data. Whichever element is smaller gets shifted to the left. After completion of one round, the largest number ends up in its correct position. In other words, the largest number bubbles to the top or right in this case. Then, the process is repeated again and again until all of the data is sorted. ***Algorithm:*** Step 1: START Step 2: Starting with the first element, compare the current element with the next element of the list. Step 3: If the current element is greater than the next element of the list, swap them. Step 4: If the current element is less than the next element, move to the next element. Repeat Step 2. Step 5: STOP **Insertion Sort:** ***Theory:*** Insertion sort is the sorting mechanism where the sorted array is built having one item at a time. The array elements are compared with each other sequentially and then arranged simultaneously in some particular order. The analogy can be understood from the style we arrange a deck of cards. This sort works on the principle of inserting an element at a particular position, hence the name Insertion Sort.

	Algorithm: Step 1: START Step 2: Consider the first element to be sorted and the rest to be unsorted Step 3: Compare with the second element: 1. If the second element < the first element, insert the element in the correct position of the sorted portion 2. Else, leave it as it is Step 4: Repeat 2 and 3 until all elements are sorted. Step 5: STOP **Merge Sort:** ***Theory:*** In Merge sort the idea is to split the unsorted list into smaller groups until there is only one element in a group. Then, group two elements in the sorted order and gradually build the size of the group. Every time the merging happens, the elements in the groups must be compared one by one and combined into a single list in the sorted order. This process continues till all the elements are merged and sorted. ***Algorithm:*** Step 1: START Step 2: Split the unsorted list into groups recursively until there is one element per group Step 3: Compare each of the elements and then group them Step 4: Repeat step 3 until the whole list is merged and sorted in the process Step 5: STOP
Complexity	**Selection Sort:** Best: $\Omega(n^2)$ Average: $\theta(n^2)$ Worst: $O(n^2)$

	Bubble Sort: Best: $\Omega(n)$ Average: $\theta(n^2)$ Worst: $O(n^2)$ **Insertion Sort:** Best: $\Omega(n)$ Average: $\theta(n^2)$ Worst: $O(n^2)$ **Merge Sort:** Best: $\Omega(n \log(n))$ Average: $\theta(n \log(n))$ Worst: $O(n \log(n))$

Program	**Selection Sort:**
	`print('Shivam Tawari Sem: 3 Roll no: A-49')`
	
	`num=int(input('Total numbers in the list:'))`
	`list1=[int(input('Enter Number:')) for x in range(num)]`
	
	`print('Unsorted List:',list1)`
	
	`for i in range(len(list1)-1):`
	`  min_value=i`
	`  for j in range(i,len(list1)):`
	`  if list1[j]<list1[i]:`
	`  min_value=j`
	`  list1[i],list1[j]=list1[j],list1[i]`
	
	`print('Sorted List:',list1)`
	
	Bubble Sort:
	`print('Shivam Tawari Sem: 3 Roll no: A-49')`
	
	`num=int(input('Total numbers in the list:'))`
	`list1=[int(input('Enter Number:')) for x in range(num)]`
	
	`print("Unsorted list:",list1)`
	
	`for j in range(len(list1)):`
	`  for i in range(0,len(list1)-1):`
	`  if list1[i]>list1[i+1]:`
	`  list1[i],list1[i+1]=list1[i+1],list1[i]`
	`  else:`
	`  print(list1)`
	`  print()`
	`print("Sorted list:",list1)`

Insertion Sort:

```
print('Shivam Tawari Sem: 3 Roll no: A-49')

num=int(input('Total numbers in the list:'))
list1=[int(input('Enter Number:')) for x in range(num)]

print("Unsorted list:",list1)

for i in range(1,len(list1)):
    current_element=list1[i]
    pos=i
    while current_element<list1[pos-1] and pos>0:
        list1[pos]=list1[pos-1]
        pos=pos-1
        list1[pos]=current_element
print("Sorted list:",list1)
```

Merge Sort:

```
print('Shivam Tawari Sem: 3 Roll no: A-49')
num=int(input('Total numbers in the list:'))
list1=[int(input('Enter Number:')) for x in range(num)]

print("Unsorted list:",list1)

def merge_sort(list1):
   if len(list1)>1:
       mid = len(list1)//2
       left_half = list1[:mid]
       right_half = list1[mid:]
       merge_sort(left_half)
       merge_sort(right_half)
       i=0
       j=0
       k=0
       while i < len(left_half) and j < len(right_half):
           if left_half[i] < right_half[j]:
               list1[k]=left_half[i]
               i=i+1
           else:
               list1[k]=right_half[j]
               j=j+1
           k=k+1

       while i < len(left_half):
           list1[k]=left_half[i]
           i=i+1
           k=k+1

       while j < len(right_half):
           list1[k]=right_half[j]
           j=j+1
           k=k+1
merge_sort(list1)
print("Sorted List:",list1)
```

Output

Selection Sort:

```
In [6]: runcell(0, 'C:/Users/shiva/untitled0.py')
Shivam Tawari Sem: 3 Roll no: A-49

Total numbers in the list:5

Enter Number:2

Enter Number:1

Enter Number:6

Enter Number:2

Enter Number:5
Unsorted List: [2, 1, 6, 2, 5]
Sorted List: [1, 2, 2, 5, 6]
```

Bubble Sort:

```
In [9]: runcell(0, 'C:/Users/shiva/Documents/GitHub/Python/Algorithms/
Algorithms/Bubble Sort.py')
Shivam Tawari Sem: 3 Roll no: A-49

Total numbers in the list:4

Enter Number:9

Enter Number:5

Enter Number:13

Enter Number:0
Unsorted list: [9, 5, 13, 0]
[5, 9, 13, 0]

[5, 9, 0, 13]
[5, 0, 9, 13]

[0, 5, 9, 13]
[0, 5, 9, 13]

[0, 5, 9, 13]
[0, 5, 9, 13]
[0, 5, 9, 13]

Sorted list: [0, 5, 9, 13]
```

Insertion Sort:

```
In [14]: runcell(0, 'C:/Users/shiva/Documents/GitHub/Python/Algorithms
Algorithms/Insertion Sort.py')
Shivam Tawari Sem: 3 Roll no: A-49

Total numbers in the list:6

Enter Number:2

Enter Number:88

Enter Number:32

Enter Number:11

Enter Number:5

Enter Number:33
Unsorted list: [2, 88, 32, 11, 5, 33]
Sorted list: [2, 5, 11, 32, 33, 88]
```

Merge Sort:

```
In [19]: runcell(0, 'C:/Users/shiva/Documents/GitHub/Python/Algorithms
Algorithms/Merge Sort.py')
Shivam Tawari Sem: 3 Roll no: A-49

Total numbers in the list:5

Enter Number:22

Enter Number:5

Enter Number:88

Enter Number:49

Enter Number:0
Unsorted list: [22, 5, 88, 49, 0]
Sorted List: [0, 5, 22, 49, 88]
```