**Information Technology**

**UIT2201 — Programming and Data Structures**

**2022 – 2023**

**Exercise — 05**

1)Arrange n elements either in ascending or descending order using Bubble sort and Selection sort. Write a Python function to sort n numbers and analyze the time complexity of your code and express the same in asymptotic notation. Give your inference on the performance of these sorting algorithms in terms of the number of comparisons and number of swaps performed. Try the best case, worst case and average case scenarios and report your observations.

2)Write a Python function to sort n numbers using Insertion sort and analyze the time complexity of your code using the number of comparisons and swaps required and express the same in asymptotic notation. Try the best case, worst case and average case scenarios and report your observations.

**AIM:**

To execute the Bubble sort, Selection sort, Insertion sort and find the number of comparsions and number of swaps.

**CODE:**

#importing the random and time module

import random

import timeit

import math

def bubble\_sort(lst):

    start\_time = timeit.default\_timer()

    '''

    This function can sort the given list by swapping the adjacent elements.

    '''

    no\_comp=0

    no\_swap=0

    n=len(lst)                                           #To find the length of list

    for i in range(0,n-1):

        for j in range(0,n-i-1):

            no\_comp+=1

            if lst[j]>lst[j+1]:

                lst[j],lst[j+1]=lst[j+1],lst[j]         #Simultaneous Swapping

                no\_swap+=1

    print("Bubble Sort: ",lst)

    print("Number of Comparsion: ",no\_comp)

    print("Number of Swap: ",no\_swap)

    print("Ratio Analysis for n: ",no\_comp/n)

    print("Ratio Analysis for n2: ",no\_comp/(n\*\*2))

    print("Ratio Analysis for n3: ",no\_comp/(n\*\*3))

    print("Ratio Analysis for log n: ",no\_comp/(math.log(n)))

    time = timeit.default\_timer() - start\_time

    print("Exact Time: ",time)

    return lst

def selection\_sort(lst1):

    start\_time = timeit.default\_timer()

    '''

    This function can sort the given list by finding the minimum value in list.

    And

    '''

    no\_comp=0

    no\_swap=0

    n=len(lst1)                                                  #To find the length of list

    for i in range(0,n):

        min\_ind=i

        for j in range(i+1,n):

            no\_comp+=1

            if lst1[j]<min\_ind:

                min\_ind=j

        no\_comp+=1

        if i!=min\_ind:

            lst1[j],lst1[min\_ind]=lst1[min\_ind],lst1[j]        #Simultaneous Swapping

            no\_swap+=1

    print("Selection Sort: ",lst1)

    print("Number of Comparsion: ",no\_comp)

    print("Number of Swap: ",no\_swap)

    print("Ratio Analysis for n: ",no\_comp/n)

    print("Ratio Analysis for n2: ",no\_comp/(n\*\*2))

    print("Ratio Analysis for n3: ",no\_comp/(n\*\*3))

    print("Ratio Analysis for log n: ",no\_comp/(math.log(n)))

    time = timeit.default\_timer() - start\_time

    print("Exact Time: ",time)

    return lst1

def insertion\_sort(lst):

    start\_time = timeit.default\_timer()

    '''

    This function sorts the given sequence in ascending order and return the sorted list.

    '''

    no\_comp = 0

    no\_swap = 0

    n=len(lst)                                               #To find the length of list

    for i in range(1, len(lst)):

        val = lst[i]

        j = i-1

        while j >= 0 and lst[j] > val:

            no\_comp += 1

            lst[j+1] = lst[j]                                 #Simultaneous Swapping

            no\_swap += 1

            j -= 1

        lst[j+1] = val

    print("Insertion Sort: ",lst)

    print("Number of Comparsion: ",no\_comp)

    print("Number of Swap: ",no\_swap)

    print("Ratio Analysis for n: ",no\_comp/n)

    print("Ratio Analysis for n2: ",no\_comp/(n\*\*2))

    print("Ratio Analysis for n3: ",no\_comp/(n\*\*3))

    print("Ratio Analysis for log n: ",no\_comp/(math.log(n)))

    time = timeit.default\_timer() - start\_time

    print("Exact Time: ",time)

    return lst1

#Creating a list

lst=[]

num=int(input("Enter Number of Elements: "))

last=int(input("Enter Last Number: "))

for i in range(num):

    a=random.randint(1,last)

    lst.append(a)                                       #adding the elements in the list

print("Orginal List: ",lst)

#Cloning

lst1=lst.copy()

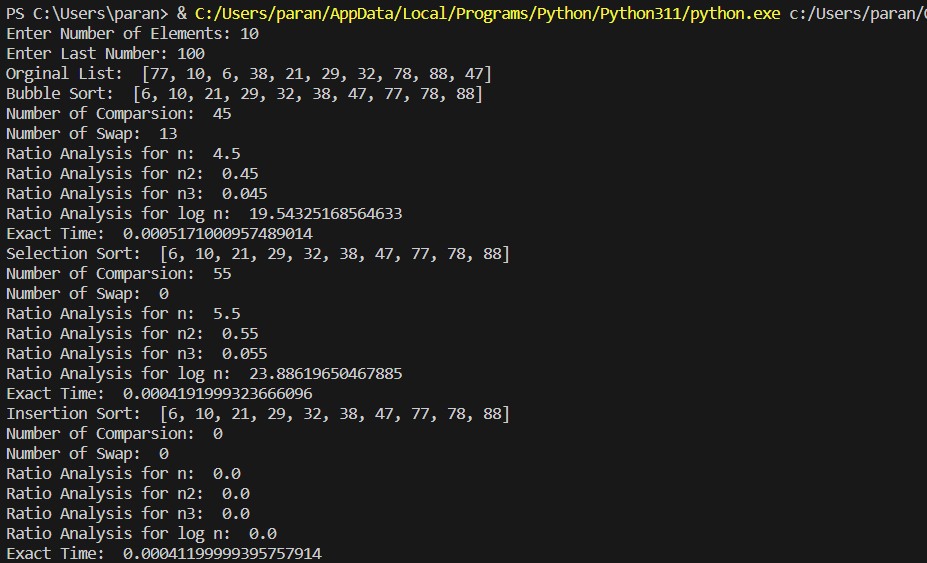
#Function Calling

bubble\_sort(lst)

selection\_sort(lst)

insertion\_sort(lst)

**OUTPUT:**

****