**Experiment No. – 1.4**

**Aim**: Write a program to implement Insertion sort along with its complexity analysis.

1. **Problem Description:**

We have given an unsorted array of numbers, generate a sorted array of numbers by applying Insertion Sort. Demonstrate knowledge of time complexity of Insertion Sort by counting the number of operations involved in each iteration and find space complexity.

1. **Algorithm:**

**Algorithm Insertion\_Sort (DATA, N):**

1. Repeat steps 2 to 4 for K = 2,3…N:
2. Set TEMP= arr[K] and J=k-1
3. Repeat while TEMP<arr[J] and J>=1
   1. Set arr[J+1]= arr[J]. [Moves element forward]
   2. Set J=J-1

[End of loop]

4. Set arr[J+1]=TEMP. [Inserts element in proper place]

[End of step 1 Outer loop]

5. Exit

1. **Complexity Analysis**

**Time complexity of Insertion Sort**

In first pass requires 1 comparison, the 2nd pass requires 2 comparisons, and kth element requires (k-1) and finally the last pass requires (n-1) comparisons. Therefore, total no. of comparisons is:

F(n) = 1+2+3….+(n-1)=n(n-1)/2 = O(n2 )

Best Case: O(n) when the input array is sorted .

Average: O(n2)

Worst Case: O(n2) when the input array are in descending order.

**Space Complexity**: O(1). As no extra space is used while sorting.

1. **Pseudo Code**

procedure insertionSort( Arr : array of items )

int Position

int value

for i = 1 to length(A) inclusive do: /\* select value to be inserted \*/

value = Arr[i]

Position = i /\*locate hole position for the element to be inserted \*/

While Position > 0 and A[holePosition-1] > value do:

A[Position] = A[Position-1]

Position =Position -1

end while

A[Position] = value /\* insert the number at hole position \*/

end for

end procedure

1. **Source Code (C/C++):**

#include <bits/stdc++.h>

using namespace std;

void insertionSort(int arr[], int n)

{

    int position, value;

    for (int i = 1; i < n; i++)

    {

        value = arr[i];

        position = i - 1;

        while (arr[position] > 0 && arr[position] > value)

        {

            arr[position + 1] = arr[position];

            position--;

        }

        arr[position + 1] = value;

    }

}

int main()

{

    cout << "Name: Saurabh Kumar \nUID: 23MAI10004\n";

    int n;

    cout << "    Insertion Sort: \n";

    cout << "Enter the Size: ";

    cin >> n;

    int arr[n];

    cout << "Enter elements:" << endl;

    for (int i = 0; i < n; i++)

    {

        cin >> arr[i];

    }

    insertionSort(arr, n);

    cout << "Array after Sorting: ";

    for (int i = 0; i < n; i++)

    {

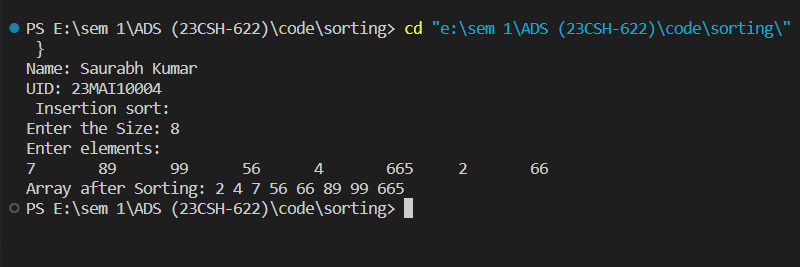
        cout << arr[i] << " ";

    }

    return 0;

}

1. **Screenshot of Outputs:**

****

1. **Learning Outcomes**
2. Learnt about how to sort an array.
3. Learnt about how to implement Insertion sort.
4. Learnt about how to find time and space complexity
5. Learnt about how to code in c++ and take array as an input.
6. Learnt about how to use function and loop in c++.