|  |  |
| --- | --- |
|  | DEPARTMENT OF ARTIFICIAL INTELLIGNECE & DATA SCIENCE |

|  |  |
| --- | --- |
| Subject: Analysis of Algorithm | Course Code: CSC402 |
| Semester: 4 | Course: AI & DS |
| Laboratory No: 205 | Name of Subject Teacher: Pramod Bhavarthe |
| Name of Student: Sahil Shaikh | Roll Id: VU2S2223012 |

|  |  |
| --- | --- |
| Title of Practical | Implement Insertion sort. |

**Theory –**

**Insertion sort** is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.

**Characteristics of Insertion Sort:**

* This algorithm is one of the simplest algorithms with simple implementation
* Basically, Insertion sort is efficient for small data values
* Insertion sort is adaptive in nature, i.e., it is appropriate for data sets which are already partially sorted.

**Time Complexity:**

* The **worst-case** time complexity of Insertion sort is **O(N^2)**
* The **average case** time complexity of Insertion sort is **O(N^2)**
* The time complexity of the **best case** is **O(N)**.

**Program –**

def insertionSort(arr):

    # Traverse through 1 to len(arr)

    for i in range(1, len(arr)):

        key = arr[i]

        # Move elements of arr[0..i-1], that are

        # greater than key, to one position ahead

        # of their current position

        j = i-1

        while j >= 0 and key < arr[j] :

                arr[j + 1] = arr[j]

                j -= 1

        arr[j + 1] = key

# Driver code to test above

arr = [12, 11, 13, 5, 6]

insertionSort(arr)

for i in range(len(arr)):

    print ("% d" % arr[i])

**Output –**

****

**Conclusion –**

**Therefore, we have successfully understood and Implemented Insertion sort.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Grade and Dated Signature of Teacher** | **Total (10)** | **Remark** | **Dated signature of teacher** |
|  |  |  |