# Vidyavardhini's College of Engineering & Technology

Department of Computer Science and Engineering (Data Science)

**ACADEMIC YEAR: 2024-25**

**Course:** Analysis of Algorithm Lab

**Course code:** CSL401

**Year/Sem:** SE/IV

| |
|---|
| **Experiment No.:** |
| **Aim:** To implement Insertion Sort and Selection Sort. |
| **Name:** SOHAM HEMENDRA RAUT |
| **Roll Number:** 24 |
| **Date of Performance:** 06/01/2025 |
| **Date of Submission:** 16/01/2025 |

**Evaluation**

| Performance Indicator | Max. Marks | Marks Obtained |
|---|---|---|
| Performance | 5 | |
| Understanding | 5 | |
| Journal work and timely submission. | 10 | |
| **Total** | **20** | |

| Performance Indicator | Exceed Expectations (EE) | Meet Expectations (ME) | Below Expectations (BE) |
|---|---|---|---|
| Performance | 5 | 3 | 2 |
| Understanding | 5 | 3 | 2 |
| Journal work and timely submission. | 10 | 8 | 4 |

**Checked by**

**Name of Faculty** : Mrs. Komal Champanerkar

**Signature** :

**Date** :

- **Aim: To implement Insertion Sort and Selection Sort.**
- **Theory:**

  Insertion sort : Insertion sort is a simple sorting algorithm that builds the final sorted array one element at a time by comparing and inserting each element into its correct position.

  Selection sort :Selection sort is a simple sorting algorithm that repeatedly selects the smallest (or largest) element from the unsorted portion of the list and swaps it with the first unsorted element until the list is sorted.

- **Algorithms:**

  Algorithm INSERTION-SORT() :

  1. Create an empty list to store the numbers.
  2. Write a function to exchange two elements in the list.
  3. Ask the user how many numbers they want to add to the list.
  4. Use a loop to take the numbers as input and add them to the list.
  5. Start from the second number in the list:
     a. Compare it with the numbers before it.
     b. If it's smaller, swap it backward until it's in the right place.
  6. Finally, print the sorted list.

  Algorithm SELECTION-SORT() :

- **Simple Algorithm for Sorting a List:**
  1. Start with an empty list.
  2. Write a function  to exchange two numbers in the list.
  3. Ask the user how many numbers they want to sort and take their input.
  4. Add the numbers to the list one by one.
  5. For each number in the list:
     a. Assume it is the smallest.
     b. Check all the numbers after it to find a smaller one.
     c. If a smaller number is found, swap it with the current number.
  6. Print the sorted list.
- **Complexity:**

  <u>Insertion Sort:</u>

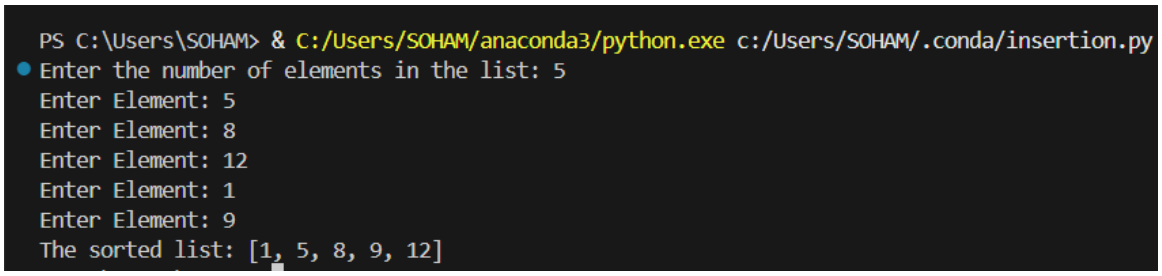
  Best Case Analysis: O(n)

Worst Case Analysis: O(n^2) Average Case Analysis: O(n^2)

Selection Sort:
Best Case Analysis: O(n^2)
Worst Case Analysis: O(n^2)
Average Case Analysis: O(n^2)

## ❖ **Program 1: Insertion Sort and output**

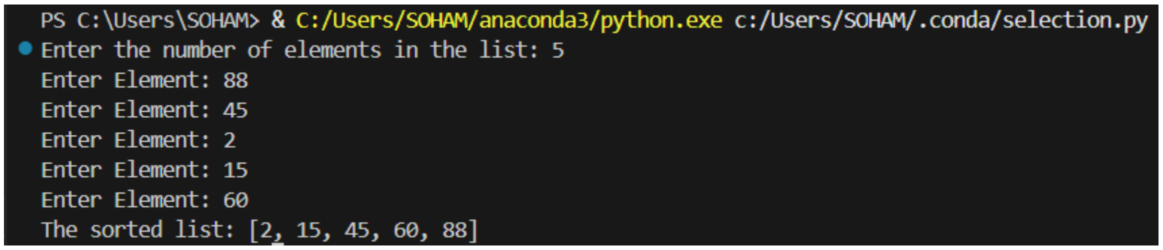
```
def swap(a, b):

   temp = lst[a]

   lst[a] = lst[b]

   lst[b] = temp

n = int(input("Enter the number of elements in the list: "))

lst = []

for i in range(n):

   element = int(input("Enter Element: "))

   lst.append(element)

# Insertion sort implementation

for i in range(1, n):

   temp = lst[i]

   temp_index = i

   for j in range(i - 1, -1, -1):

      if temp < lst[j]:
            swap(temp_index, j)

            temp_index -= 1

print("The sorted list:", lst)
```



```
PS C:\Users\SOHAM> & C:/Users/SOHAM/anaconda3/python.exe c:/Users/SOHAM/.conda/insertion.py
Enter the number of elements in the list: 5
Enter Element: 5
Enter Element: 8
Enter Element: 12
Enter Element: 1
Enter Element: 9
The sorted list: [1, 5, 8, 9, 12]
```

## ❖ **<u>Program 2: Selection Sort and output</u>**

```
def swap(a, b):
    temp = lst[a]
    lst[a] = lst[b]
    lst[b] = temp
n = int(input("Enter the number of elements in the list: "))
lst = []
for i in range(n):
    element = int(input("Enter Element: "))
    lst.append(element)
# Insertion sort implementation
for i in range(1, n):
    temp = lst[i]
    temp_index = i
    for j in range(i - 1, -1, -1):
        if temp < lst[j]:
            swap(temp_index, j)
            temp_index -= 1
    print("The sorted list:", lst)
```



```
PS C:\Users\SOHAM> & C:/Users/SOHAM/anaconda3/python.exe c:/Users/SOHAM/.conda/selection.py
Enter the number of elements in the list: 5
Enter Element: 88
Enter Element: 45
Enter Element: 2
Enter Element: 15
Enter Element: 60
The sorted list: [2, 15, 45, 60, 88]
```

❖ **Conclusion:** Selection sort and insertion sort are simple sorting algorithms, with insertion sort being faster for nearly sorted data, while both have O(n^2) time complexity for average and worst cases.