**PRACTICAL-9**

**AIM:** Implement a Python Program to perform the following Sorting and Searching algorithms:

i) Linear search

ii) Binary search

iii) Selection sort

iv) Insertion sort

**Source Code:**

edef linear\_search(arr, target):

for i in range(len(arr)):

if arr[i] == target:

return i

return -1

def binary\_search(arr, target):

left, right = 0, len(arr) - 1

while left <= right:

mid = (left + right) // 2

if arr[mid] == target:

return mid

elif arr[mid] < target:

left = mid + 1

else:

right = mid - 1

return -1

def selection\_sort(arr):

for i in range(len(arr)):

min\_idx = i

for j in range(i+1, len(arr)):

if arr[j] < arr[min\_idx]:

min\_idx = j

arr[i], arr[min\_idx] = arr[min\_idx], arr[i]

def insertion\_sort(arr):

for i in range(1, len(arr)):

key = arr[i]

j = i - 1

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

arr[j + 1] = arr[j]

j -= 1

arr[j + 1] = key

arr = [64, 34, 25, 12, 22, 11, 90]

target = 25

linear\_result = linear\_search(arr, target)

print(f"Linear Search: {target} found at index {linear\_result}" if linear\_result != -1 else f"{target} not found")

arr.sort()

binary\_result = binary\_search(arr, target)

print(f"Binary Search: {target} found at index {binary\_result}" if binary\_result != -1 else f"{target} not found")

selection\_sort(arr.copy())

print("Selection Sort:", arr)

insertion\_sort(arr.copy())

print("Insertion Sort:", arr)

**Output:**

