

ALGORITHM ANALYSIS AND DESIGN

PRACTICAL -6

Create an application for Ganpat University to list and sort student's marks. Repeat the experiment for different values of n. The number of elements in the list to be sorted and plot a graph of the comparison count of any 3 sorting algorithms.

CODE:

```
import random
import matplotlib.pyplot as plt
import numpy as np

def insertion_sort(arr):
    n = len(arr)
    comparisons = 0
    for i in range(1, n):
        key = arr[i]
        j = i - 1
        #
        while j >= 0 and arr[j] > key:
            arr[j + 1] = arr[j]
            j -= 1
        comparisons += 1
        #
        arr[j + 1] = key
    print(arr)
    return comparisons

def quick_sort(arr):
    if len(arr) <= 1:
        return 0

    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]

    left_comparisons = quick_sort(left)
    right_comparisons = quick_sort(right)
```

```
total_comparisons = left_comparisons + right_comparisons + len(arr) -
len(left) - len(right)

return total_comparisons


def selection_sort(arr):
    cmp = 0
    for a in range(len(arr)):
        min_index = a
        for b in range(a + 1, len(arr)):
            cmp += 1
            if arr[b] < arr[min_index]:
                # cmp+=1
                min_index = b
        arr[a], arr[min_index] = arr[min_index], arr[a]
    return cmp

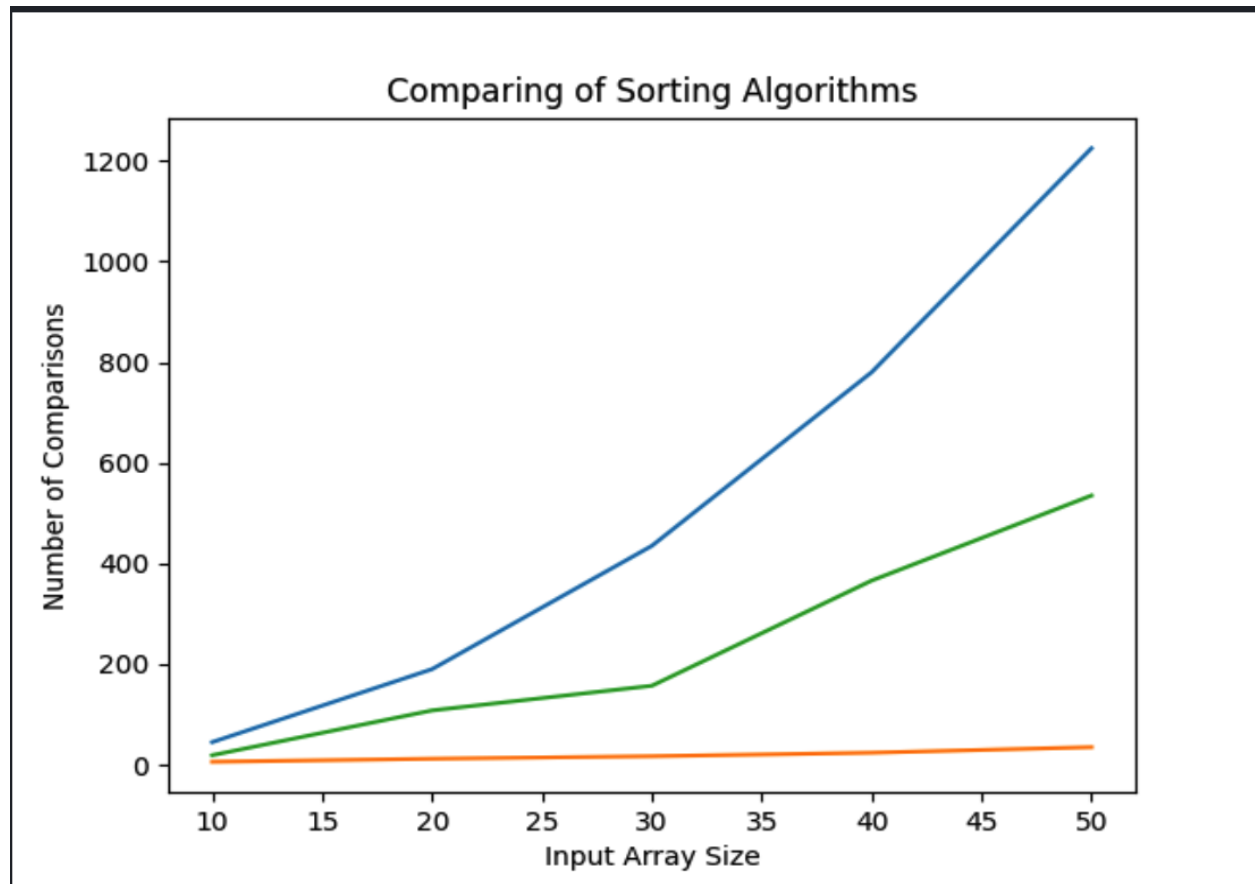
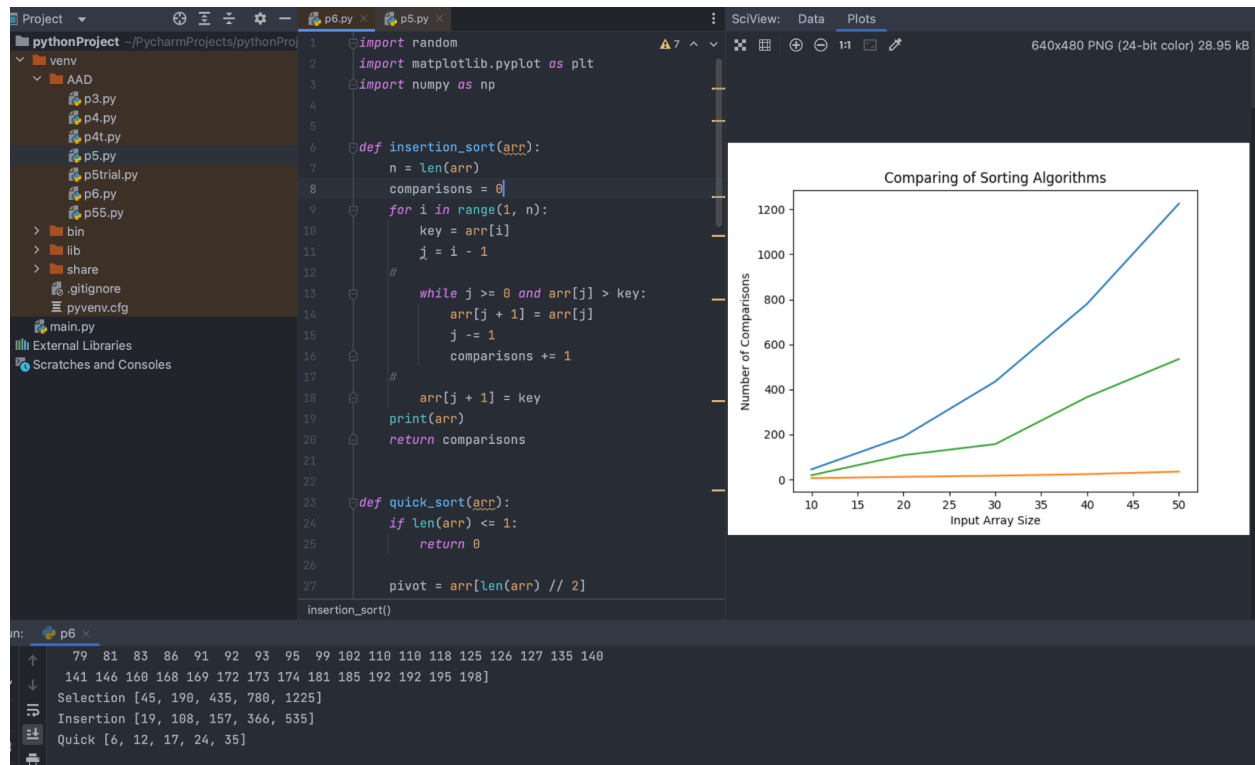

arr = [10, 20, 30, 40, 50]
selection = []
insertion = []
quick = []
for j in range(len(arr)):
    nums = np.random.randint(1, 200, arr[j])
    insertion.append(insertion_sort(nums))

    quick.append(quick_sort(nums))

    selection.append(selection_sort(nums))


print("Selection", selection)
print("Insertion", insertion)
print("Quick", quick)


plt.plot(arr, selection)
plt.plot(arr, quick)
plt.plot(arr, insertion)
plt.ylabel("Number of Comparisons")
plt.xlabel("Input Array Size")
plt.title("Comparing of Sorting Algorithms")
plt.show()
```

OUTPUT:

KARIYA RAJ

21162101011