**30-8-2023**

**LAB EXERCISE 1**

**SORTING**

**ANANYA R**

**CB.EN.U4CSE21006**

**SAJITH RAJAN**

**CB.EN.U4CSE21052**

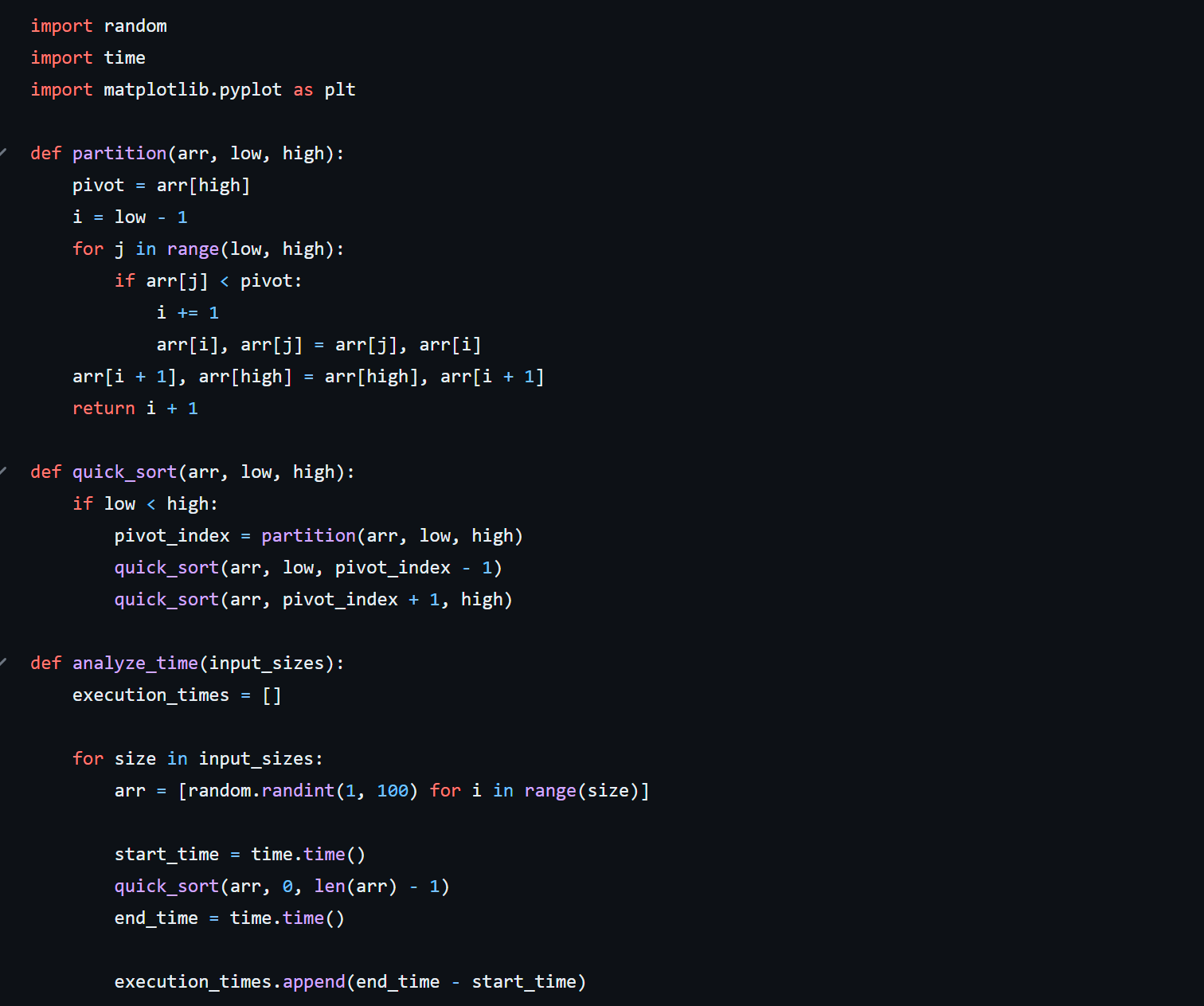
**19CSE302**

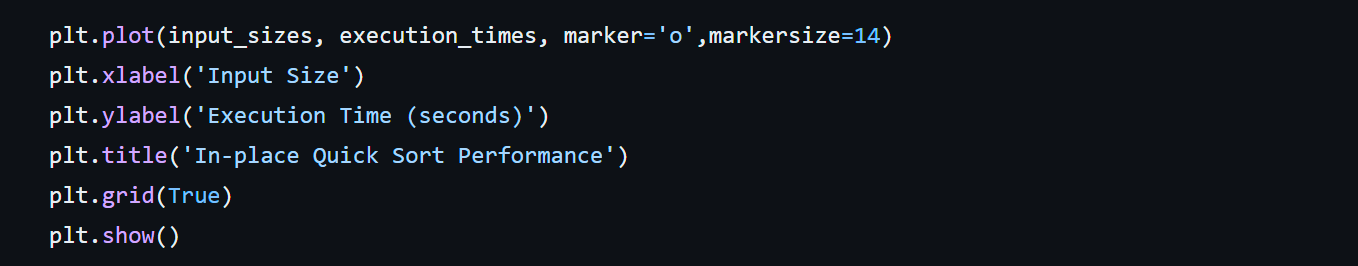
**LAB ASSIGNMENT PROGRESS**

**Question 1.**

In the first part implement the basic sorting algorithms 1. In-place Quick Sort 2. Merge Sort 3. In-place Heap Sort 4. Insertion Sort 5. Bucket Sort 6. Radix Sort. The pseudocode for a few algorithms is attached and the heap ADT is also given. Your implementation should follow the pseudo-code and use the ADT for heapsort. Generate multiple random input test cases of various sizes (e.g. 100 (min), 500, 1000) and evaluate the different algorithms based on a. Number of comparisons (if applicable) b. Number of swaps (if applicable) c. Number of basic operations (other than above) d. Running time in milliseconds e. Memory used Based on the above provide your observations and analysis.

**In place quick sort**

****

****

*# Bucket Sort in Python*

def bucketSort(*array*):

    bucket = []

*# Create empty buckets*

*for* i *in* range(len(*array*)):

        bucket.append([])

*# Insert elements into their respective buckets*

*for* j *in* *array*:

        index\_b = int(10 \* j)

        bucket[index\_b].append(j)

*# Sort the elements of each bucket*

*for* i *in* range(len(*array*)):

        bucket[i] = sorted(bucket[i])

*# Get the sorted elements*

    k = 0

*for* i *in* range(len(*array*)):

*for* j *in* range(len(bucket[i])):

*array*[k] = bucket[i][j]

            k += 1

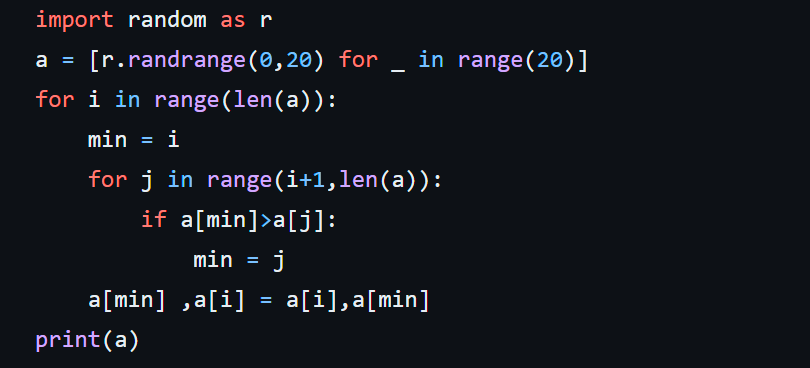
*return* *array*

array = [.42, .32, .33, .52, .37, .47, .51]

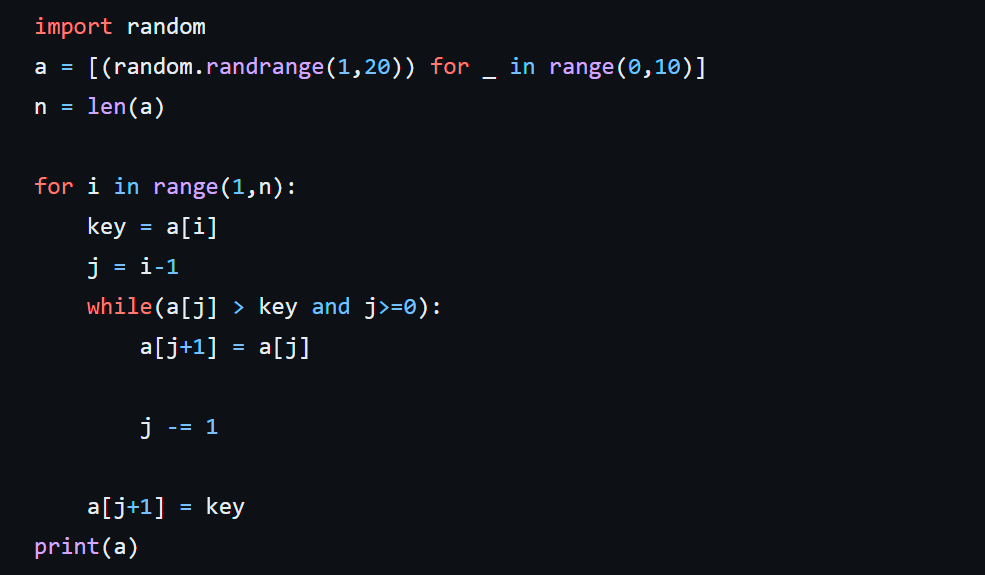
print("Sorted Array in descending order is")

print(bucketSort(array))

**Selection Sort**

****

**Insertion Sort**

****

**RADIX SORT:**

*# Radix sort in Python*

*# Using counting sort to sort the elements in the basis of significant places*

def countingSort(*array*, *place*):

    size = len(*array*)

    output = [0] \* size

    count = [0] \* 10

*# Calculate count of elements*

*for* i *in* range(0, size):

        index = *array*[i] // *place*

        count[index % 10] += 1

*# Calculate cumulative count*

*for* i *in* range(1, 10):

        count[i] += count[i - 1]

*# Place the elements in sorted order*

    i = size - 1

*while* i >= 0:

        index = *array*[i] // *place*

        output[count[index % 10] - 1] = *array*[i]

        count[index % 10] -= 1

        i -= 1

*for* i *in* range(0, size):

*array*[i] = output[i]

*# Main function to implement radix sort*

def radixSort(*array*):

*# Get maximum element*

    max\_element = max(*array*)

*# Apply counting sort to sort elements based on place value.*

    place = 1

*while* max\_element // place > 0:

        countingSort(*array*, place)

        place \*= 10

data = [121, 432, 564, 23, 1, 45, 788]

radixSort(data)

print(data)