ACADEMY OF TECHNOLOGY

Lab Assignment (Day 5)

Paper name: Design and Analysis of Algorithms Lab

Code: PCC-CS494 **Semester: 4^{th}**

Discipline: CSE **Time: 2 Hours**

Date: March 28, 2023

1. Write a program in C or C++ to sort a given array of elements using the QuickSort algorithm, applying the partition algorithm as follows:
   1. Hoare's partition algorithm
   2. Lomuto's partition algorithm
   3. Implement Randomized Quick Sort algorithm.

Algorithm:

Algorithm 1: **QuickSort** ***(arr,low,high)***

Input: An array $arr[low : high]$ is a global array to be sorted.
Output: Sorted array such that $arr[i] \leq arr[i+1]$ for all $1 \leq i \leq n$

```
// Here low ⟹ Starting index and high ⟹ Ending index
if low < high then
    // j is partitioning index, arr[j] is now at right place
    j := PARTITION(arr, low, high);
    QUICKSORT(arr, low, j − 1); // Before arr[j]
    QUICKSORT(arr, j + 1, high); // after arr[j]
end
```

Algorithm 2: **Hoare-Partition-Left** *(arr,low,high)*

```
   // This function takes first element as pivot, places the pivot
      element at its correct position in sorted array, and places
      all smaller (smaller than pivot) to left of pivot and all
      greater elements to right of pivot
 1 i := low;
 2 j := high + 1;
 3 pivot := arr[low];
 4 while i < j do
 5 |   do i := i + 1; while (i ≤ j and arr[i] < pivot);
 6 |   do j := j − 1; while (arr[j] > pivot);
 7 |   if i < j then Interchange(arr, i, j)
 8 end
 9 Interchange(arr, low, j);
10 return j;
```

Algorithm 3: **Lomuto-Partition-Right** *(arr,low,high)*

```
   // This function takes last element as pivot, places the pivot
      element at its correct position in sorted array, and places
      all smaller (smaller than pivot) to left of pivot and all
      greater elements to right of pivot
 1 pivot := arr[high];
 2 i := low − 1; // Temporary pivot index
 3 for j := low to high − 1 do
   |   // If the current element is less than or equal to the pivot
 4 |   if arr[j] ≤ pivot then
 5 |   |   i := i + 1; // Move the temporary pivot index forward
 6 |   |   Interchange (arr, i, j); // Swap the current element with
   |   |      the element at the temporary pivot index
 7 |   end
 8 end
   // Move the pivot element to the correct pivot position
      (between the smaller and larger elements
 9 i := i + 1;
10 Interchange (arr, i, high);
11 return i;
   // the pivot index
```