Experiment 4 5th April 2022

**Quick Sort**

Aim: To write a program to implement quick sort using Divide and Conquer.

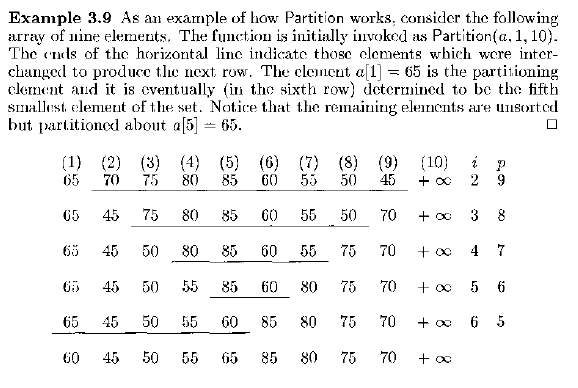
Theory:

Quick Sort was given by C. A. R. Hoare in 1962. Quick sort is a very fast sorting technique. It is an in-place sort so no additional space is required for sorting. It is also known as partition exchange sort and is very efficient because the exchanges occur between elements that are far apart, and so less exchanges are needed to place an element in its final position.

Choose an element from the list and place it at its proper position in the list i.e. at the position where it would be in the final sorted list. We call this element as **pivot** and it will be at its proper place if:

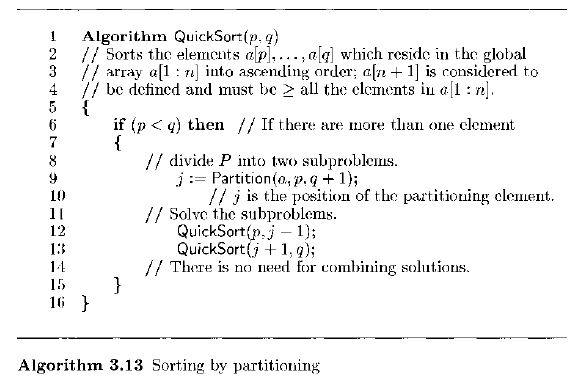
1. All the elements to the left of pivot are less than or equal to the pivot,
2. All the elements to the right of pivot are greater than or equal to the pivot.

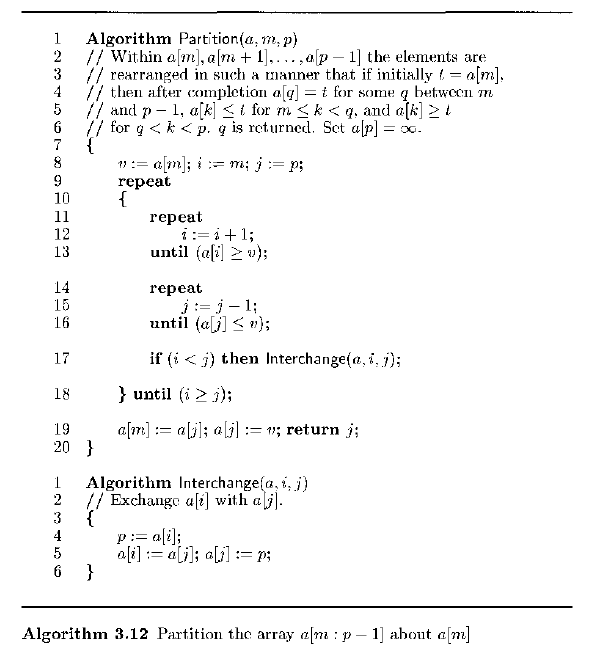
Any element is taken as pivot but for convenience the first element is taken as the pivot. The list is partitioned in such a way that pivot comes at its proper place. This is accomplished by rearranging the elements in A[1:n] such that A[i]<=A[j] for all ‘i’ between 1 and m and all ‘j’ between m+1 and ‘n’ for some m, 1<=m<=n. The left sub list is sorted using quick sort recursively, then the right sub list is sorted using quick sort recursively. The division into two sub arrays is made so that the sorted sub-arrays do not need to be merged later.



Task:

Write a program for the following algorithm:





Trace the algorithm for the following set of values showing i, j and pivot for each step: 65, 70, 80, 85, 60, 55, 50, 45

Program and Output:

Conclusion: