QUICK SORT ALGORITHM

I/ DEFINITION:

- **Quick sort** is a highly efficient sorting algorithm and is based on partitioning of array of data into smaller arrays. A large array is partitioned into two arrays one of which holds values smaller than the specified value, say pivot, based on which the partition is made and another array holds values greater than the pivot value.

- **Quick sort** partitions an array and then calls itself recursively twice to sort the two resulting subarrays. This algorithm is quite efficient for large-sized data sets as its average and worst case complexity are of O(n^2), where **n** is the number of items.

==

II/ QUICKSORT PIVOT ALGORITHM:

Step 1: Choose the highest index value has pivot

Step 2: Take 2 variables to point left and right of the list excluding pivot

Step 3: Left points to the low index

Step 4: Right points to the high

Step 5: While value at left is less than pivot move right

Step 6: While value at right is greater than pivot move left

Step 7: If both step 5 and 6 does not match swap left and right.

Step 8: If left ≥ right, the point where they met is new pivot

==

III/ QUICKSORT ALGORITHM:

Step 1: Make the right most index value pivot

Step 2: Partition the array using pivot value

Step 3: Quicksort left partition recusively

Step 4: Quicksort right partition recusively

==

III/ QUICKSORT PSEUDOCODE:

- To get more into it, let see the pseudocode for quick sort algorithm
–

```
procedure quickSort(left, right)
  if right-left <= 0
    return
  else
    pivot = A[right]
    partition = partitionFunc(left, right, pivot)
    quickSort(left,partition-1)
    quickSort(partition+1,right)
  end if
end procedure
```