Quick Sort

Quick sort is a highly efficient sorting algorithm and based on partitioning of array of data into smaller arrays.

A large array is partitioned into two arrays, one which holds values less than some specified value called pivot and other holds values greater that pivot.

Quick sort partitions array and calls itself twice to recursively two resulting sub arrays.

Algorithm –

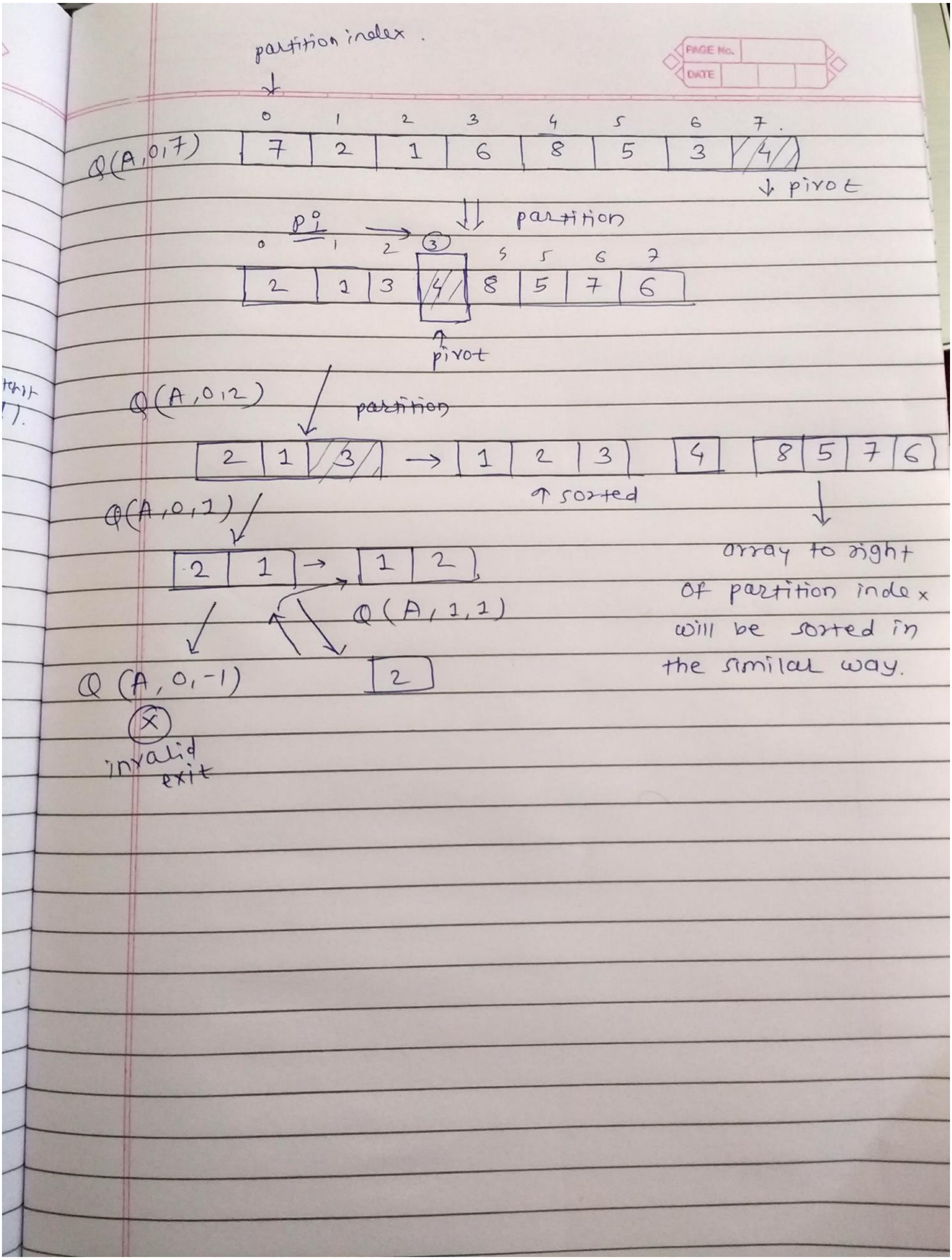
There will be two procedures one for partitioning and the other for sorting.

```
Procedure mergeSort(arr[],start,end)
{
        Base condition:
        If(start>=end)
        return arr;

        Partition index=partition(arr,start,end)             //call partition to get the partition index
        mergeSort(arr,0,partition index - 1 , end)           //recursively sort left sub array
        mergeSort(arr,start, partition index +1)             //recursively sort right sub array
        return arr;
}
Procedure partition(arr,start,end)
{
        Pivot=arr[end]                                       //make the last element of the array pivot.
        Partition_index=0                                    //Initially keep partition index 0
        For(i=start to end)
        {
                If(arr[i]<=pivot)
                        Swap arr[i] and arr[pivot]
                        Increment partition_index
        }
        After we get the partition index swap pivot and the value at partition index
        Swap arr[patition_index] and pivot

        Return Partition_index
}
```

Working explained with Example



array to right of partition index will be sorted in the similar way.

Merge Sort vs Quick Sort

- Worst case complexity of merge sort- O(nlogn). But it is not in-place sorting algorithm (in-place sorting takes constant amount of extra memory).
- Space complexity O(n)   (Extra memory required is proportional to the number of elements in the array).
- On the other hand, average case complexity of quick sort is O(nlogn) but it takes almost constant space and is in-place sorting algorithm.