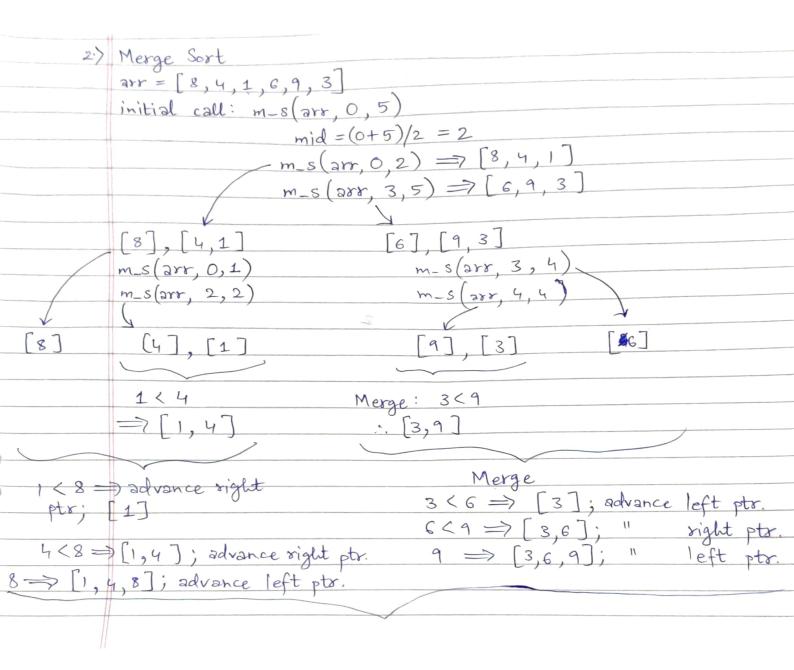
```
Classenate

Data.
Page
```

```
Exp. 2A
1) Quick Sort
    arr = [8,4,1,6,9,3]
    Initial call: quick_sort(arr, 0, 5)
    Exec. of partition (arr, 0, 5):
            pivot = arr[0] = 8
             i = low + 1 = 0 + 1 = 1
            For j=1, arr[1] = 4 < 8 \Rightarrow swap(arr[i], arr[j]
             :: i = j \implies no \text{ change}; itt \implies i = 2
            For j=2, arr[2]=1<8 \Rightarrow swap; i=j \Rightarrow no change,
            For j=3, arr[3]=6(8)
            For ;=4, arr [4] = 9 > 8 => no swap => i remains 4.
            For j=5, arr [5]=3 < 8 => swap(arr[4], arr [5])
             \Rightarrow arr = [8, 4, 1, 6, 3, 9] \Rightarrow i = 5
            Final swap => swap (arr[low=0], arr[i-1=4])
             => arr = [ 3, 4, 1, 6, 8, 9] => return i-1=4
                                                           new pivot
  Recursive Calls: quick-sort ( arr, 0, 3)
                      quick -sort (arr, 5,5)
                                right subarray [9], single element
                                  implies it's already sorted.
 exec. of quick-sort (arr, 0, 4):
           partition (arr, 0, 5):
                pivot = arr[o] = 3
                For j=1, arr[1] = 4 > 3 => no swap; i remains 1.
                For j=2, arr [2] = 1 < 3 => swap (arr [17, arr [2])
                => arr = [3, 1, 4, 6, 8, 9] => i++ => i=2
                For j=3, arr[3] = 6 > 3 => no swap, i remains 2
```

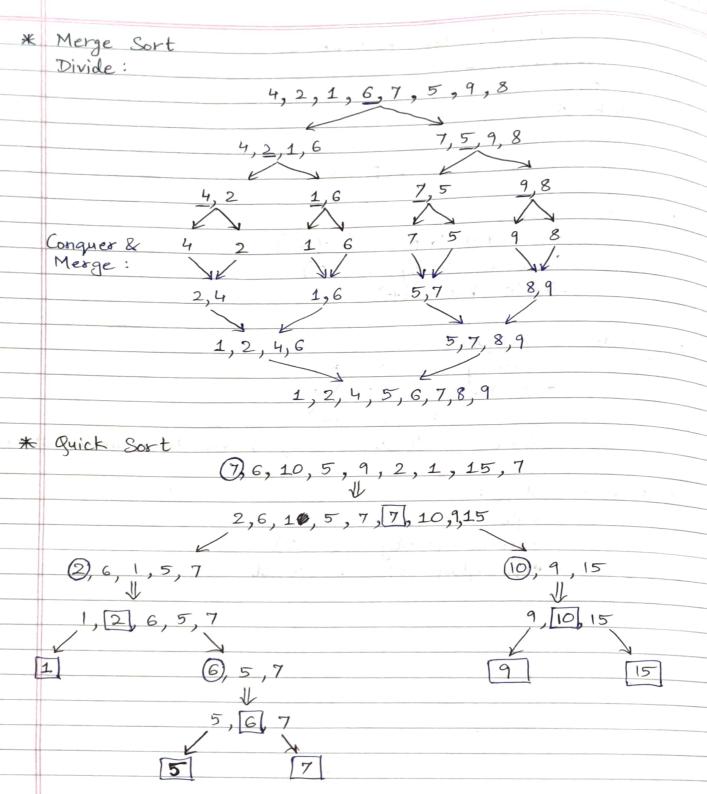
```
Final swap: Swap(arr[o], arr[i-1])

\Rightarrow \text{ arr} = [1,3,4,6,8,9] \Rightarrow \text{ return pivot-index} = 1
Recursive calls: qs(arr,0,0) \Rightarrow \text{ single element}; \text{ base case}
qs(arr,2,3) \Rightarrow \text{ right subarray} [4,6]
exec. of qs(arr,2,3):
partition(arr,2,3):
pivot = 4
i=3
For j=3: arr[3] = 6 > 4 \Rightarrow \text{ no swap} \Rightarrow i \text{ remains } 3
Final swap: (arr[2], arr[i-1]) \Rightarrow \text{ no change}
return pivot index = 2
recursive calls: <math>qs(arr,2,1) \Rightarrow \text{ empty}; \text{ base case}
qs(arr,3,3) \Rightarrow \text{ single el; } "
\therefore \text{ Final sorted array} = [1,3,4,6,8,9]
```



 $1(3 \Rightarrow [1], adv. left ptr.$ $3 < 4 \Rightarrow [1,3], " right"$ $4 < 6 \Rightarrow [1,3,4], " left ptr.$ $6 < 8 \Rightarrow [1,34,6], " right"$ $8 < 9 \Rightarrow [1,3,4,6,8], " left "$ $9 \Longrightarrow [1,3,4,6,8,9]$





```
· Master's Method for solving recurrences.
    T(n) = aT\left(\frac{n}{b}\right) + f(n) where a \ge 1, b \ge 1 & f(n) \ge 0
   Idea: Compare f(n) with n tog (a)
      Case 1: If f(n) = O(n^{(\log_b a) - \varepsilon}) for some \varepsilon > 0
then: T(n) = O(n^{\log_b a})
     Case 2: If f(n) = O(n^{\log_b^2}) then T(n) = O(n^{\log_b^2} \cdot \lg n)
     Case 3: If f(h) = \Omega(h^{(\log_b a)} + E) for some E > 0 & if
                  af(n/b) \leq cf(n) for some c < 1 & all
                   sufficiently large n then T(n) = O(f(n))
  eg. T(n) = 2T(n/2) +n
      a=2, b=2, \log 2 = 1
       => Case 2: f(n) = O(n)
           T(n) = O(n \lg n)
```