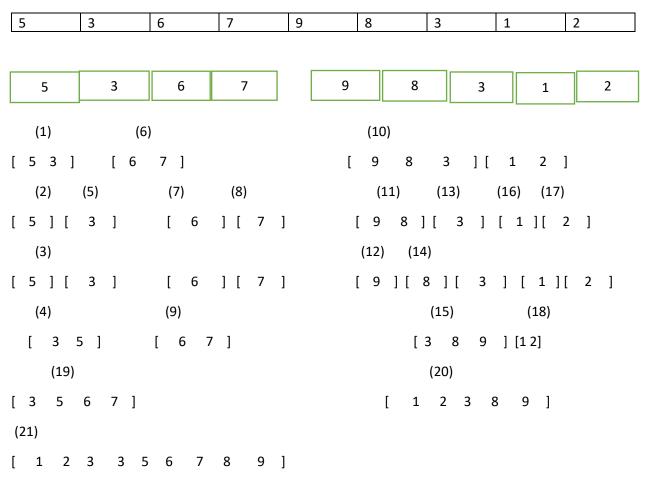
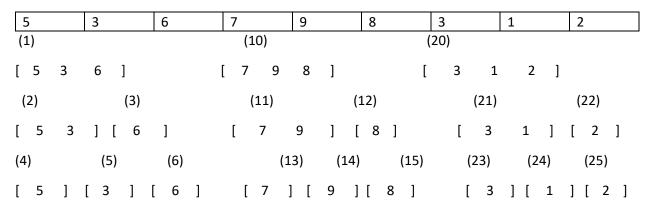
Que 17

Merge Sort comparison <u>.</u>



## Merge sort => divide into 3=>



```
(7)
               (8)
                                  (16)
                                               (17)
                                                             (26)
                                                                        (27)
[ 3
       5 ] [ 6 ]
                              [ 7
                                      9 ] [ 8 ]
                                                           [ 1
                                                                  3 ] [ 2 ]
(9)
                             (18)
                                                             (28)
[ 3
     5
         6]
                          [ 7 8
                                     9
                                        ]
                                                            [ 1 2
                                                                      3 ]
(19)
                                                (29)
[ 3
          6
               7
                     8
                         9 ]
                                             [
                                                1
                                                     2
                                                         3 ]
(30)
   1
       2
           3
               3
                   5 6
                           7 8
                                       ]
```

Time Complexity Analysis=>

2-way Merge sort we get the equation: T(n) = 2T(n/2) + O(n)Similarly, in case of 3-way Merge sort we get the equation: T(n) = 3T(n/3) + O(n)

Master Theorem=>

For 2 way merge sort, a= 2, b= 2, c= log (to the base b)a, c= 1,k=0

Run time complexity=> O(n log n)

For 3 way merge sort, a= 3, b= 3, c=log(to the base b)a

Run time complexity is  $\Rightarrow$  O(n log  $_3$ n)

By solving it using Master Theorem, we get its complexity as **O(n log 3n).**. Although time complexity looks less compared to 2 way merge sort, the time taken actually may become higher because number of comparisons in merge function go higher

## **Conclusion:**

- Are the two approaches tied?
- =>No
- If one of the approaches is better, which one is better? why?

=>I think, 2 way merge sort is better. Although time complexity looks less compared to 2 way merge sort, the time taken actually may become higher for 3 way merge sort because number of comparisons in merge function go higher.