Name :Abimanyu M

Batch :2022 9615

Enrolment NO: EBEON0323767152

Merge Sort Algorithm

MERGE\_SORT(arr, beg, end)

if beg < end

set mid = (beg + end)/2

MERGE\_SORT(arr, beg, mid)

MERGE\_SORT(arr, mid + 1, end)

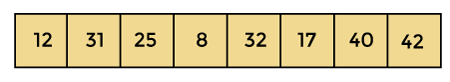
MERGE (arr, beg, mid, end)

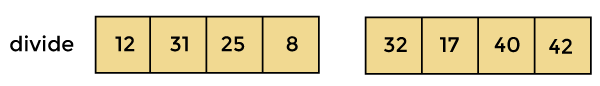
end of if

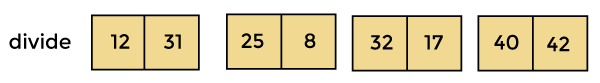
## END MERGE\_SORT

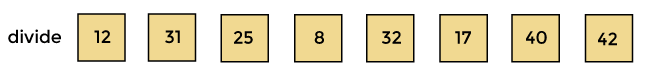
## Working of Merge sort Algorithm

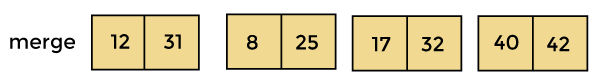
Given Array :

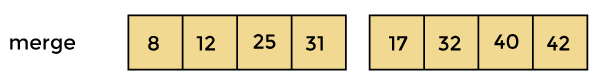
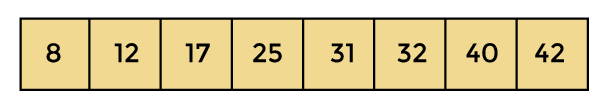












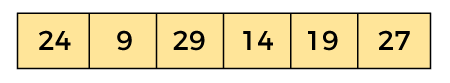
# Now, the array is completely sorted.

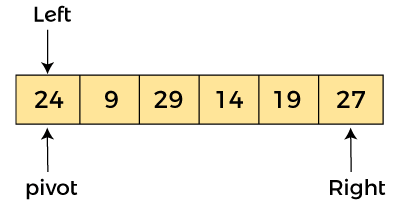
Quick Sort Algorithm

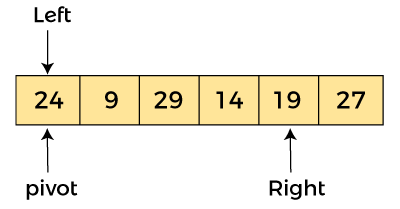
1. QUICKSORT (array A, start, end)
2. {
3. 1 **if** (start < end)
4. 2 {
5. 3 p = partition(A, start, end)
6. 4 QUICKSORT (A, start, p - 1)
7. 5 QUICKSORT (A, p + 1, end)
8. 6 }
9. }

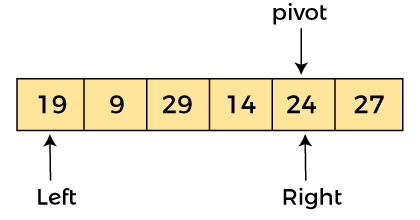
# 

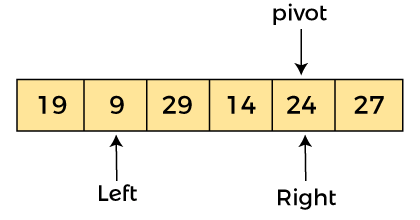
**Divide:** In Divide, first pick a pivot element. After that, partition or rearrange the array into two sub-arrays such that each element in the left sub-array is less than or equal to the pivot element and each element in the right sub-array is larger than the pivot element.

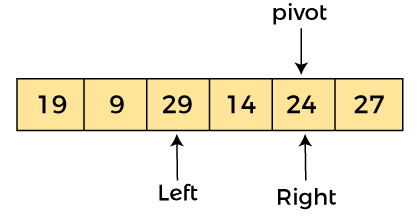
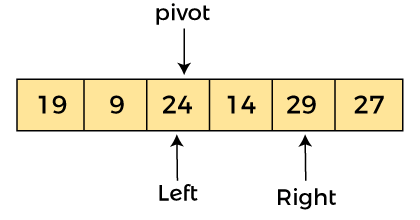
**Conquer:** Recursively, sort two subarrays with Quicksort.

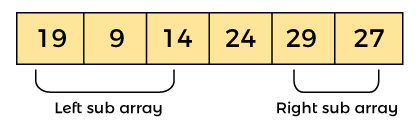
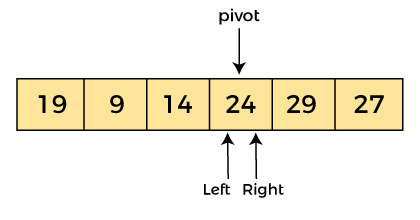
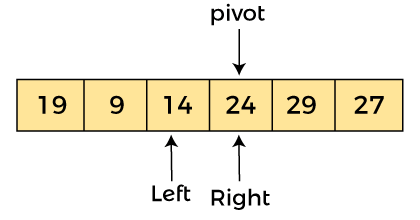
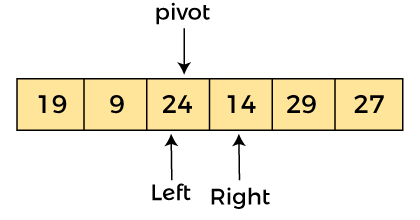












After sorting gets done, the array will be