

Merge Sort:

Concept:

Given a seq of n elements split into 2 sets $(a[1] \dots a(\frac{n}{2}))$, $[a[\frac{n}{2}+1] \dots a[n]]$

Alg:

Alg merge sort ($A[0 \dots n-1]$).

// Sorts array $A[0 \dots n-1]$ by recursive procedure.

// I/P: An array ($A[0 \dots n-1]$) of elements.

// O/P: Array $A[0 \dots n-1]$ sorted in non-decreasing order.

```
if ( $n < 1$ )  
    return;  
else {
```

Copy A $[0 \dots (\frac{n}{2}) - 1]$ to B $[0 \dots (\frac{n}{2}) - 1]$

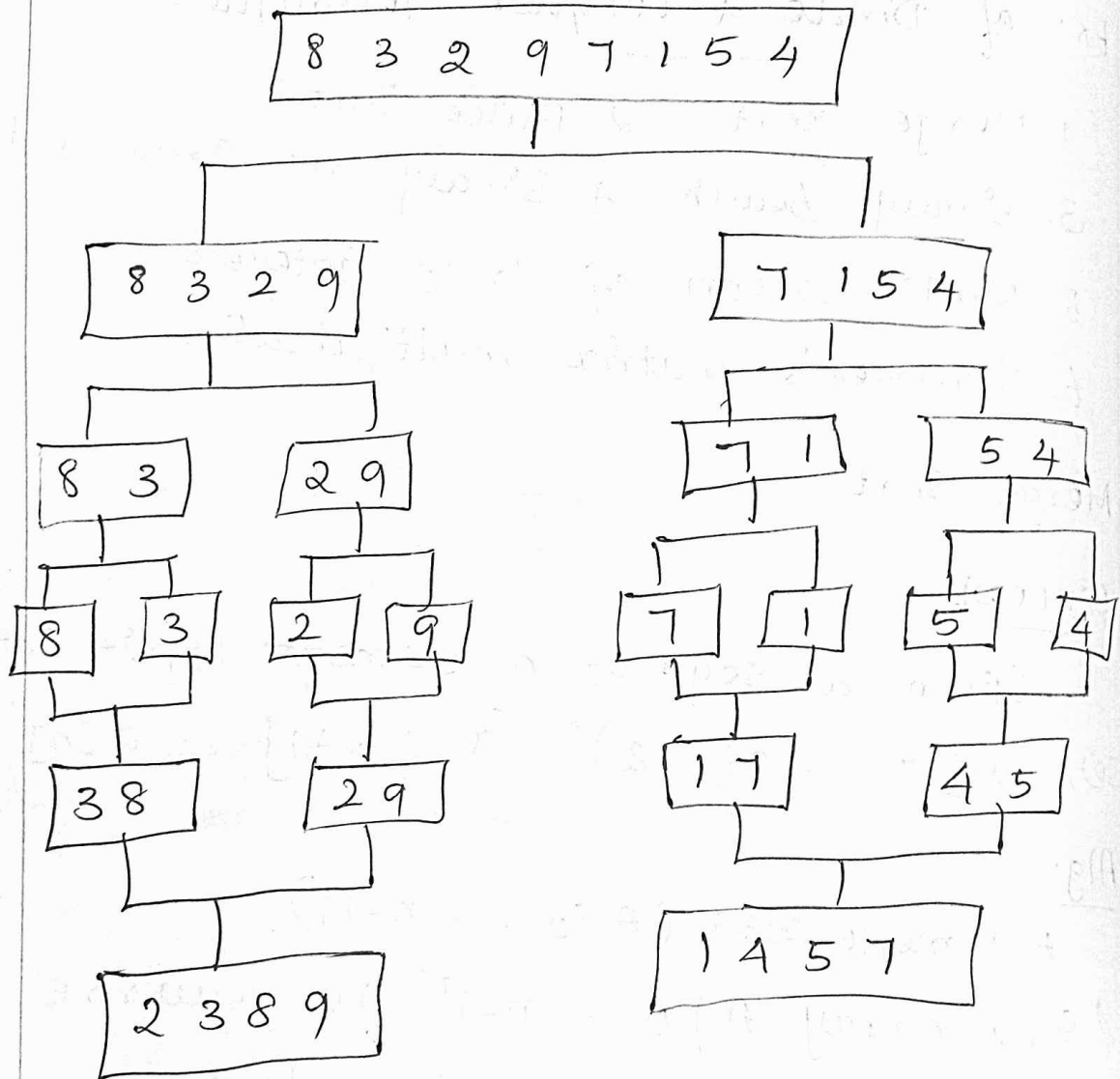
Copy A $[(\frac{n}{2}) \dots n - 1]$ to C $[0 \dots (\frac{n}{2}) - 1]$

Merge Sort $[B [0 \dots (\frac{n}{2}) - 1]]$

" $[C [0 \dots (\frac{n}{2}) - 1]]$

Merge (B, C, A)

}



Complexity for merge sort:

Complexity of an alg measured in terms of space & Time.

$$O(n \log_2 n)$$