21-09 Merge Sort - 1 }

21 September 2022 20:11

2] Easy
- 1. Insertion
- 2. Selection
- 3. Bubble

Advanced
- 1. merge [1. 2.]
- 2. Quick [1. 2.]

---

1. Insertion — worst - $O(n^2)$, Best - $\Omega(n)$
2. Selection — worst = $O(n^2)$, Best - $\Omega(n)$
3. Bubble — worst - $O(n^2)$, Best - $\Omega(n^2)$

for( )
{
O(n) [
}

---

* Advanced Algo:-

1. Merge Sort:-

a
b
c

"Merge two sorted Arrays":-

Q-1 Two arrays a1 & a2

Input:- a1 = [2, 4, 5, 6] // Sorted
a2 = [1 3 4 8] // Sorted

→ Merge these two sorted arrays

O/P:- [1 2 3 4 4 5 6 8]

Approach-1

length of a1 = m
length of a2 = n

1. Create an empty array of size m+n. // O(1)

2	4	5	6	1	3	4	8

$O(m+n)$

2. copy elements from a1 & a2 into new array.
3. Apply any known sorting algo. to sort the new array. ⇒ $O((m+n)^2)$

$$O(1) + O(m+n) + O((m+n)^2)$$

$$\Rightarrow O((m+n)^2)$$

Approach-2 Sorted

m+n

$a_1 = [\overset{P_1}{2}\ 4\ 5\ 6]$ // m
$a_2 = [\underset{P_2}{1}\ 3\ 4\ 8]$ // n

Create an empty array of size m+n

res:- [ ]
k

if (a1[P1] < a2[P2])] if it is true
2 < 1 xx ⇒ in res[k] => 1st pos of res
a1[P1]

else // a1[P1] > a2[P2]
res[k] = a2[P2]

```
if (a1[P1] < a2[P2]) {
    res[k] = a1[P1]
    P1++;
    k++;
}
else{
    res[k] = a2[P2]
    P2++;
    k++;
}
```

P1 P2

Heart

Sorted P1

Sorted

$a_1 = [2\ 4\ 5\ 6]$ // m

$a_2 = [1\ 3\ 4\ 8]$ // n

p_1 p_2

res: [1 2]

k

4 ③

3

Sorted

if

else

$a_1 = [2\ 4\ 5\ 6]$ // m

$a_2 = [1\ 3\ 4\ 8]$ // n

p_1 p_2

4<4

res: [1 2 3 4 4 5 6 8]

$a_1 = [2\ 4\ 5\ 6]$

$a_2 = [1\ 3\ 4\ 8\ 9\ 10]$ // Sorted

p_1 p_2

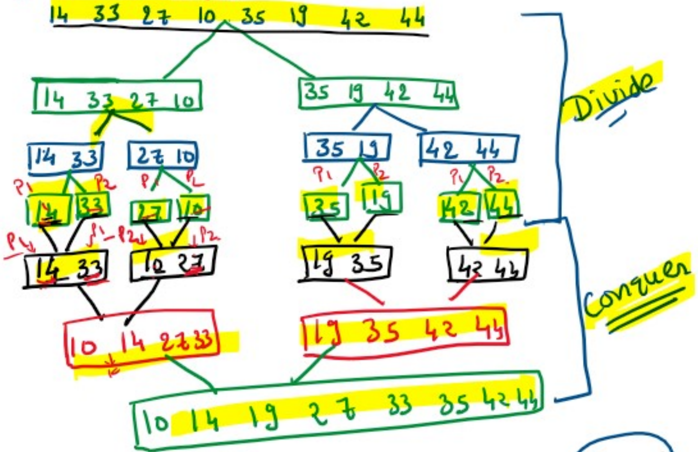
// sorted

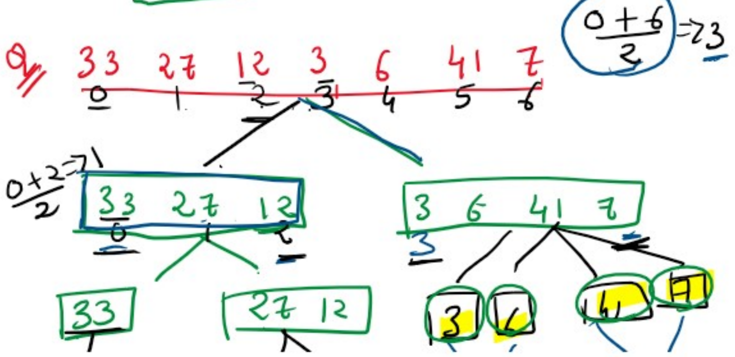
[1 2 3 4 4 5 6 8 9 10]

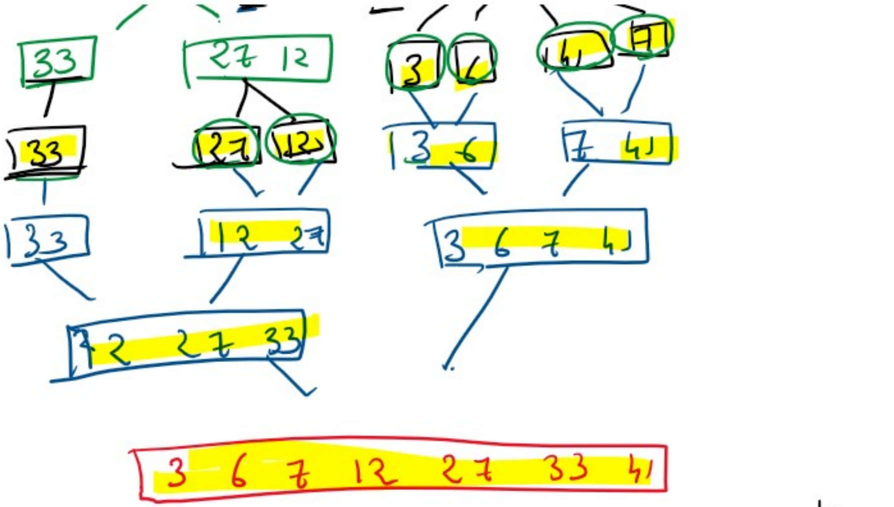
⊛ Merge Sort:-

1) Divide and Conquer.

① Divide & conquer

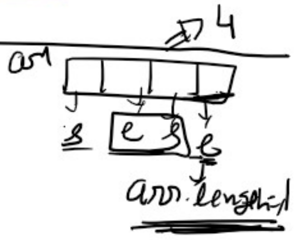


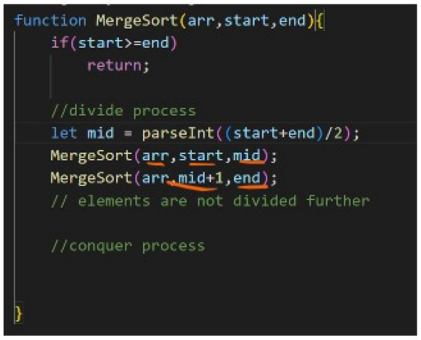




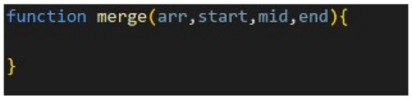
function mergeSort(arr, s, e)
{
if(start >= end)
return;
Base case

var mid = parseInt((start+end)/2)
mergeSort (arr, Start, mid)
mergeSort (arr, mid+1, end)





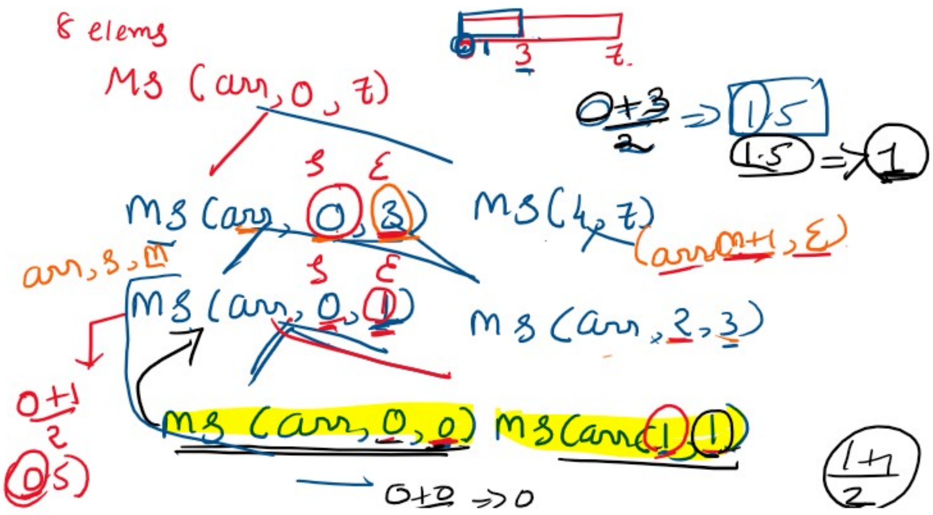
```
function MergeSort(arr,start,end){
    if(start>=end)
        return;

    //divide process
    let mid = parseInt((start+end)/2);
    MergeSort(arr,start,mid);
    MergeSort(arr,mid+1,end);
    // elements are not divided further

    //conquer process


}
```

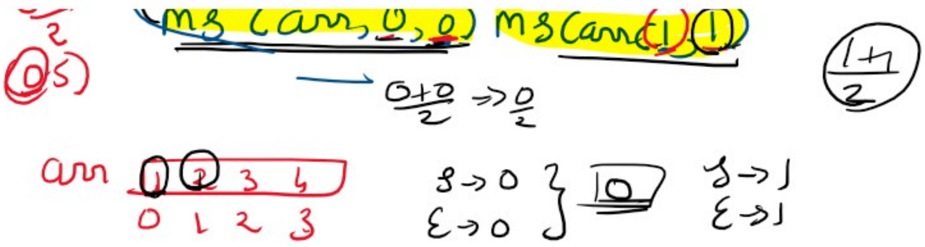


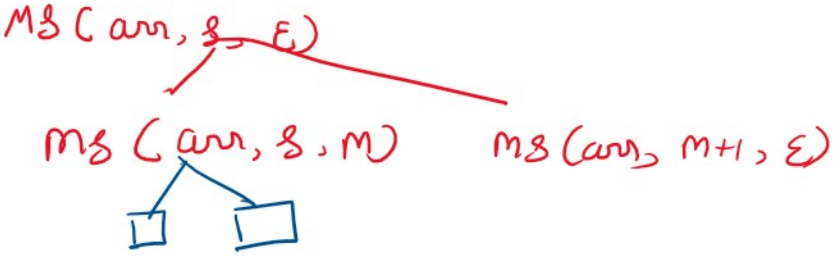
```
function merge(arr,start,mid,end){

}
```

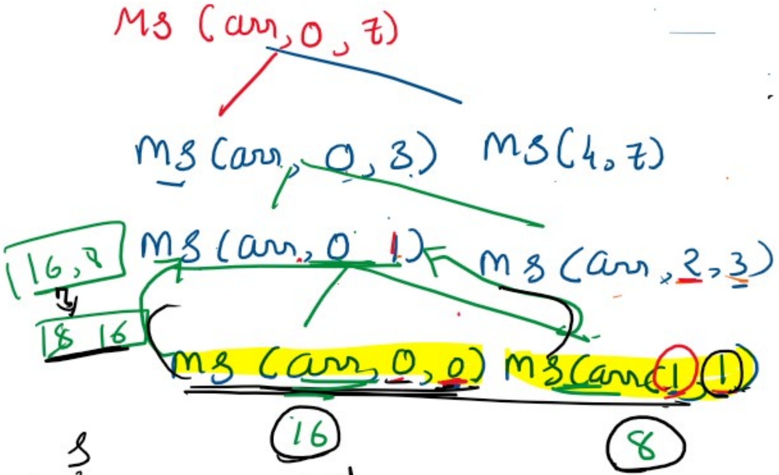
arr, s, m   arr, m+1, E

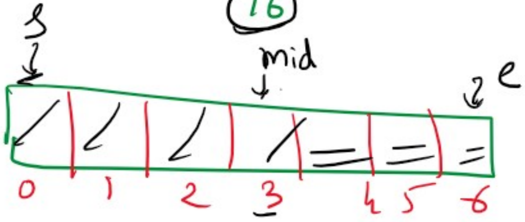
MergeTwoSortedArrays(arr, Start, mid, end)











$\frac{0+6}{2} \Rightarrow 3$

$a_1 = mid - s + 1 = 3 - 0 + 1 \Rightarrow \boxed{4} \Rightarrow a_1$

$a_2 = e - mid = 6 - 3 = \boxed{3} \Rightarrow a_2$

m1 => 4

start + i

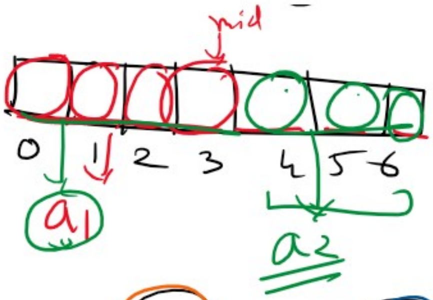
0 + 0

0 + 1

0 + 2

0 + 3

2



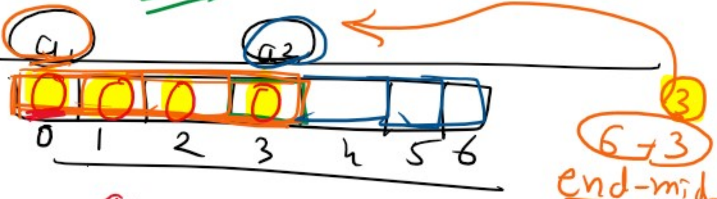
mid+1 +j

mid+1+0

mid+1+1

mid+1+3.

arr =>



6 - 3

end-mid

Start+i

0 +0

0+1

0+2

0+3

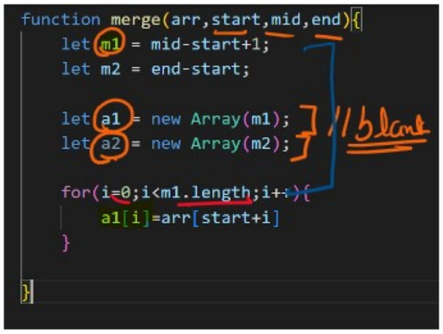
a1

$\frac{0+6}{2} \Rightarrow 3$

mid-start+1

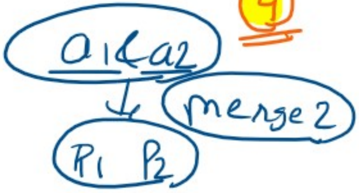
3-0+1

4



```
function merge(arr,start,mid,end){
    let m1 = mid-start+1;
    let m2 = end-start;

    let a1 = new Array(m1);
    let a2 = new Array(m2);

    for(i=0;i<m1.length;i++){
        a1[i]=arr[start+i]
    }

}
```

//blank



n [① MS (first half) (arr, s, m)

② MS (Second half) (arr, m+1, e)

③ Merge ← $\left(\frac{n}{2}, \frac{n}{2}\right)$

R.R

$$T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + O(n)$$

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n) \quad — ①$$

$$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{2^2}\right) + \frac{n}{2} \quad — ②$$

② -> ①

$$T(n) = 2\left[2T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right] + n$$

$$T(n) = 2\left[2T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right] + n \Rightarrow 2n$$

$T(n) = 2\left[2T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right] + n \Rightarrow$

$T\left(\frac{n}{2^2}\right) = 2T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}$

$T(n) = 2^3 T\left(\frac{n}{2^3}\right) + 3n \rightarrow 3$

$T(n) = 2^k T\left(\frac{n}{2^k}\right) + kn$

$T(1)$

$\frac{n}{2^k} = 1$

$n = 2^k$

$\log n = k \log_2 2$

$\boxed{k = \log n}$

$2^{\log_2 n} T(1) + n\log n$

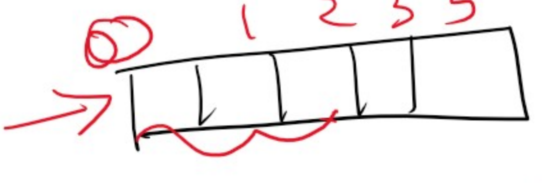
$n \times 1 + n \log n$

$\Rightarrow \boxed{n \log n}$

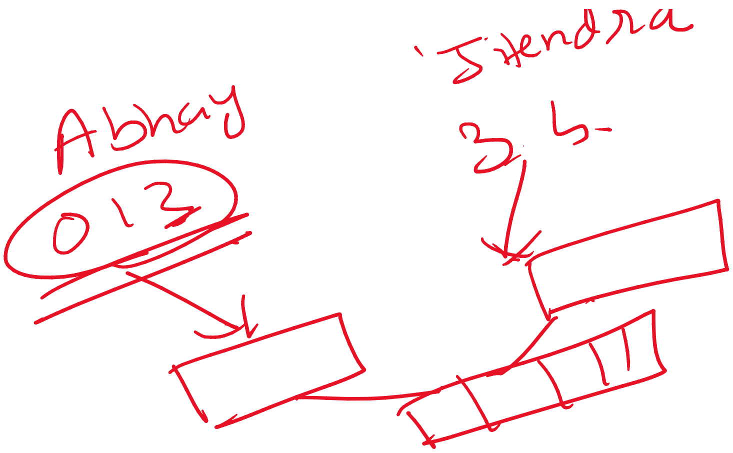
S.C $\Rightarrow O(n)$

Time $\Rightarrow O(n \log n) \rightarrow$ B W A

SC $\Rightarrow O(n)$



Jitendra


Abhay
0 1 2
Jitendra
3 5