Ques Merge 2 sorted arrays

n_1

n_2

Res

$n_1 + n_2 = n$

Create new array $(n_1 + n_2)$

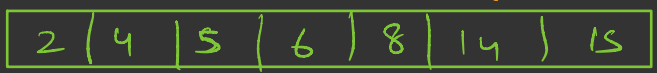
Copy both the array $(n_1 + n_2) \Rightarrow O(n)$

Sort the resultant array $\Rightarrow$ $n \log n$

TC $\Rightarrow n + n\log(n) \Rightarrow O(n \log n)$

i

j



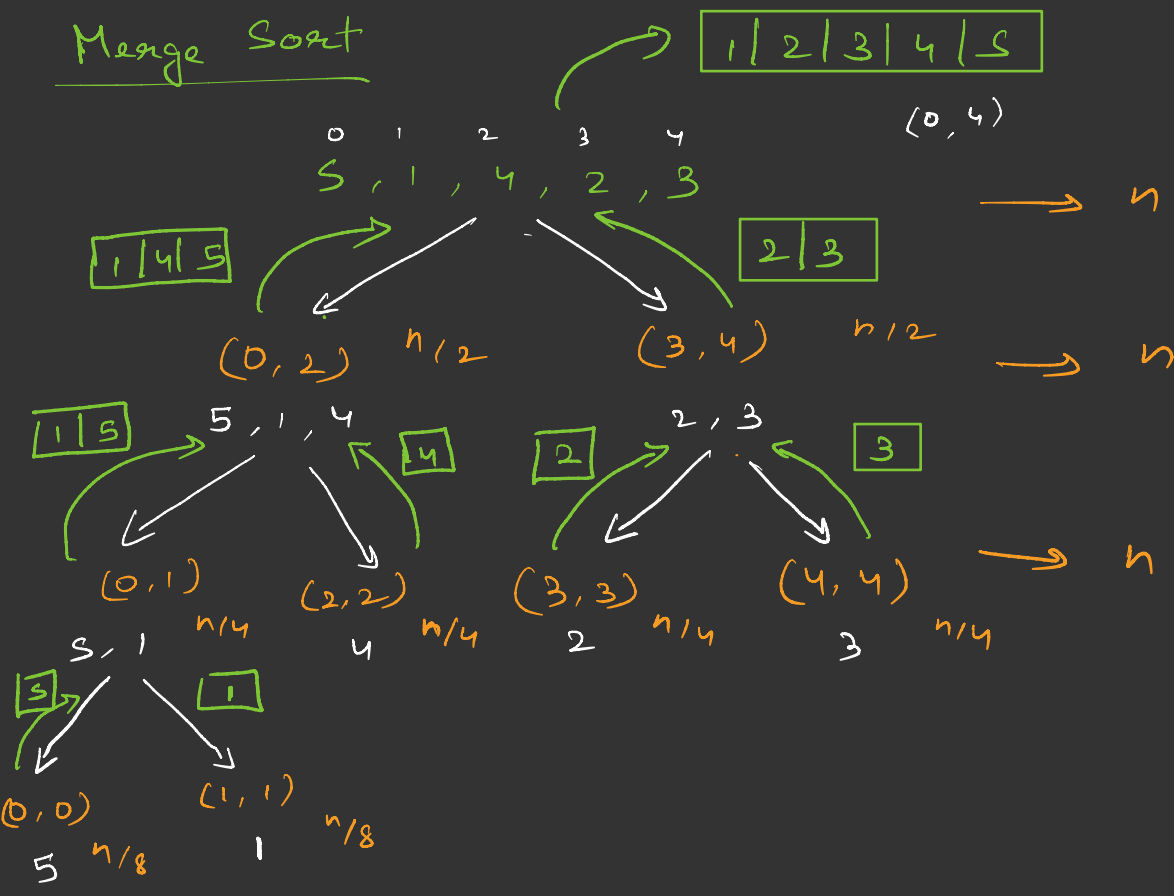
Res

k

TC $\Rightarrow O(n)$

$n = n_1 + n_2$

Merge Sort



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

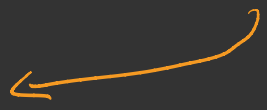
$$k \Rightarrow \log_2^{(n)}$$

TC $\Rightarrow$ no of operations per level $\times$ no of levels

$$n \times \log_2 n$$

TC $\Rightarrow O(n \log n)$

Space => Auxilary Space => O(n)



not inplace

Stable sorting algo

Ques Partition your array 4

5	1	2	4	7	3	6

<=

all elements on left of pivot should be smaller or equal to pivot

all elements on right of pivot should be greater than pivot

3	1	2	4	7	5	6

pivot = 4

1	2	4	3	7	5	6
0	1	2	3	4	5	6

i

j

① 0 to j - 1 <= pivot

② j to i - 1 > pivot

③ i to end unknown

① swap (i, j)
i++
j++

② i++

③

0 1 4 3 0 0 1 3 1 0

0 0 0 0 1 3 4 3 1 3 1

Ques Sort 0, 1

partition odd even

Ques

0 1 3 0 1 4 2 1 0

i

j

==0 0 -> j-1

!=0 j -> i-1

unknown i -> end

(1) swap(i, j)

i++

j++

(2) i++

i =✗ 3

j =✗ 2

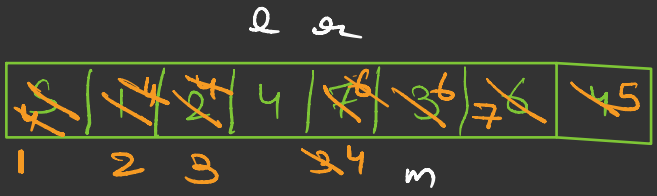
3 Way Partitioning

Sort 0, 1, 2

Ques

Partition p=4

pivot = 4



< pivot	0 to l-1	(1)
= pivot	l to m-1	(2)
Unknown	m to r	
> pivot	r+1 to end	(3)

1 2 4 5

(1)
swap(m, l)
m++
l++

(2)
m++

(3)
swap(m, r)
r--;