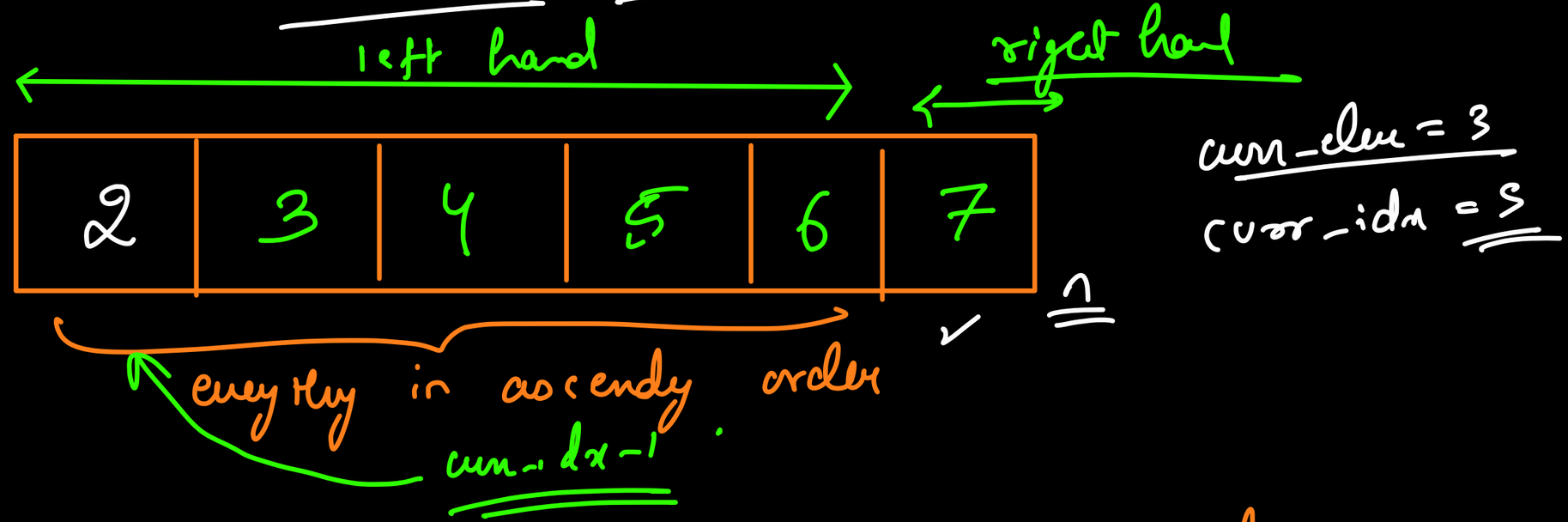
Insertion Sort

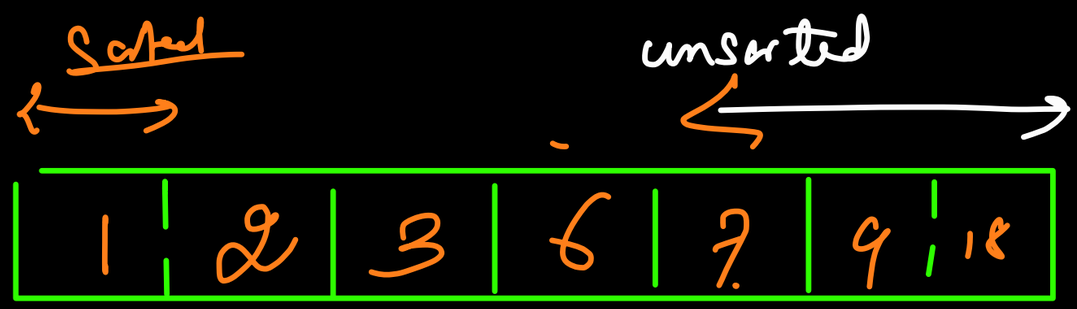


Situation

1) You've an array which has all the elements uptill the 2nd last index ($(n-2)^{th}$ index) arranged in asc order.

2) Only the last element is not present at it's correct place.

```
let temp = arr[i]
arr[i] = arr[j]
arr[j] = temp
```



$i \in [1, n-1]$

$i = 4 \quad 5 \quad 6$

curr_elem := 2

$j = i-1 \rightarrow 3 \quad 4 \quad 5 \quad 6$

```
while( arr[j] > curr_elem) {
        arr[j+1] = arr[j];
        j--;
}
arr[j+1] = curr_elem
```

Space → $O(1)$

Time → $O(n^2)$ $\quad \Omega(n)$

$1 + 2 + 3 + \cdots\cdots\cdots$

[3, 4, 5, 2, 1]

what if the array is V Sorted ?
already / almost

[1, 2, 3, 4, 5, 6]

[1, 3, 4, 5, 6, 2]

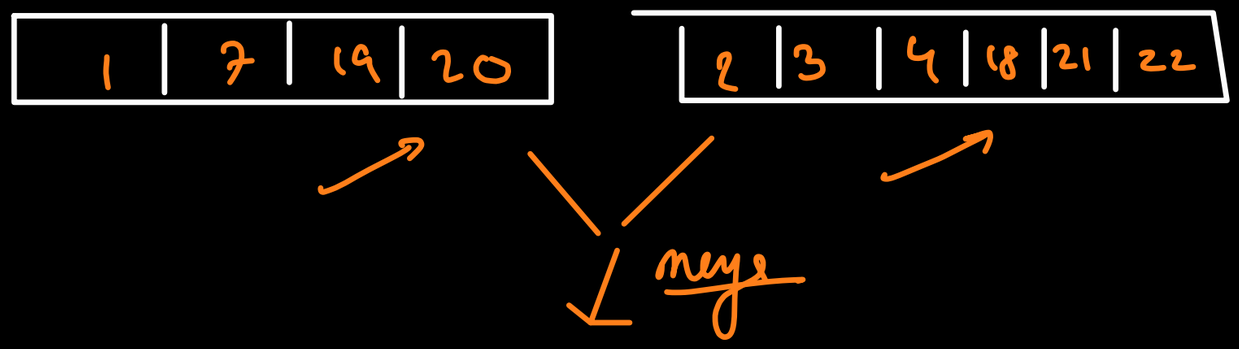
In place → Yes

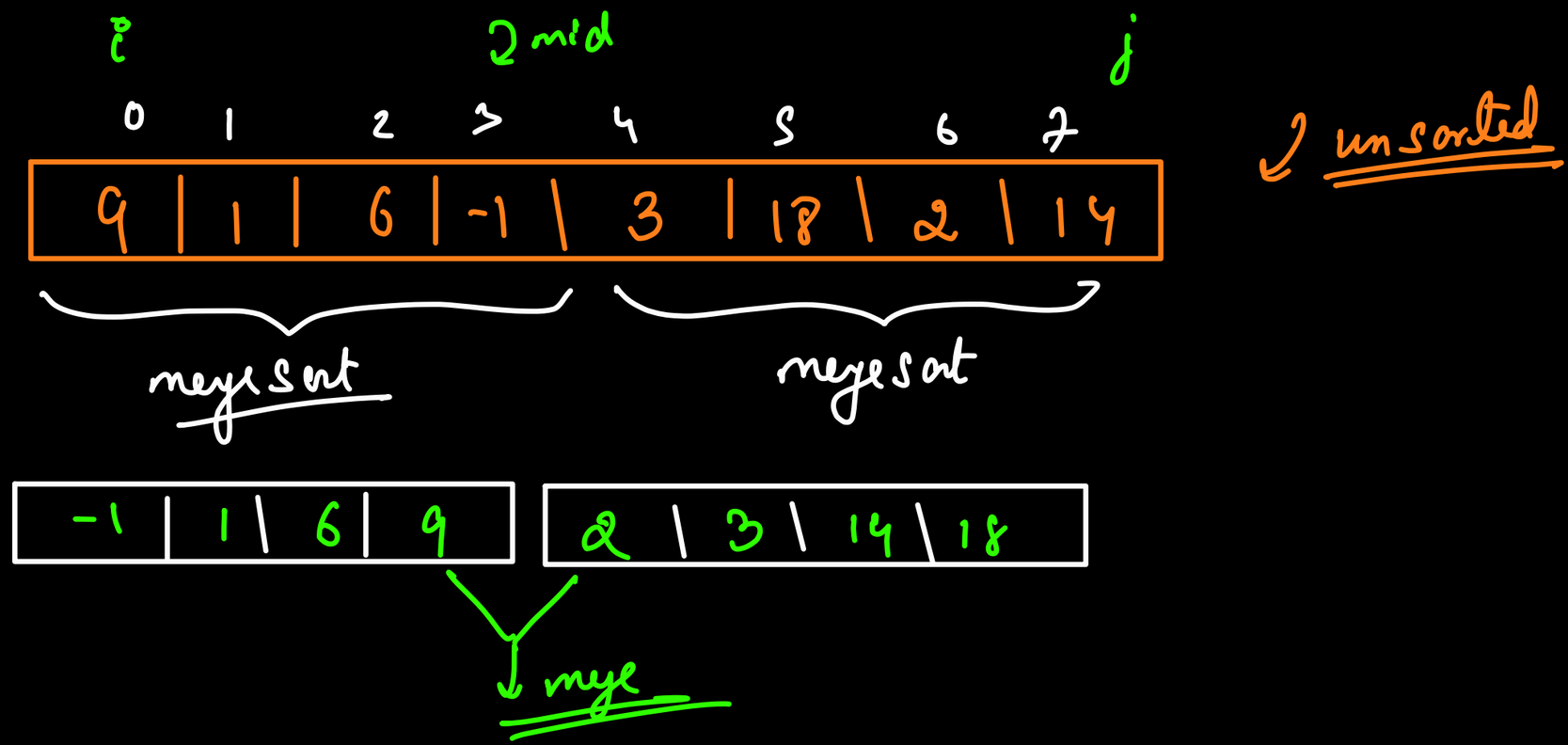
Stability ?? → Yes

1, 2, ~~3~~2, ~~2~~3

curr-ele : 2

Merge Sort





$f(arr, i, j)$ → $f(arr, i, mid)$

↓

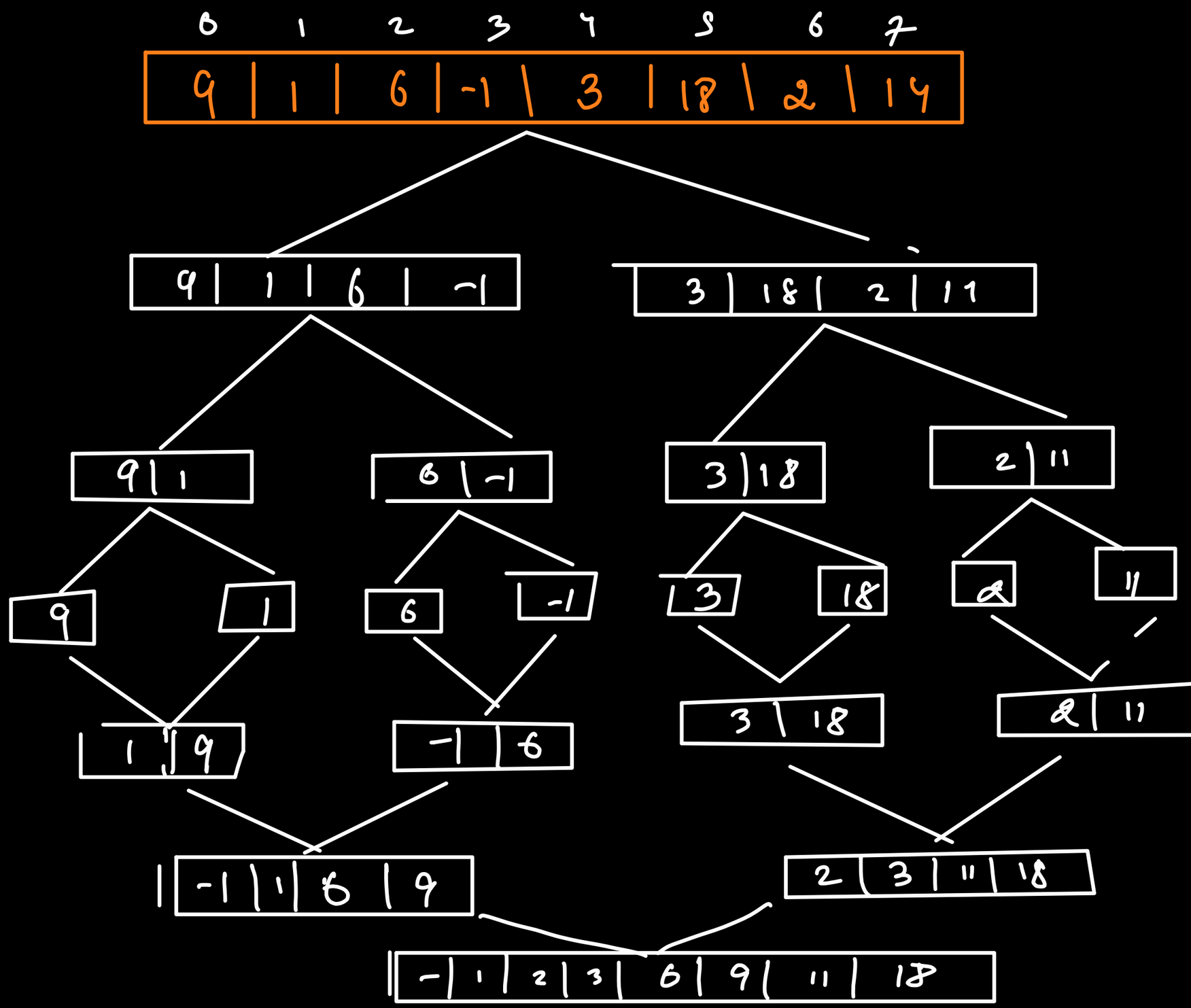
$f(arr, mid+1, j)$

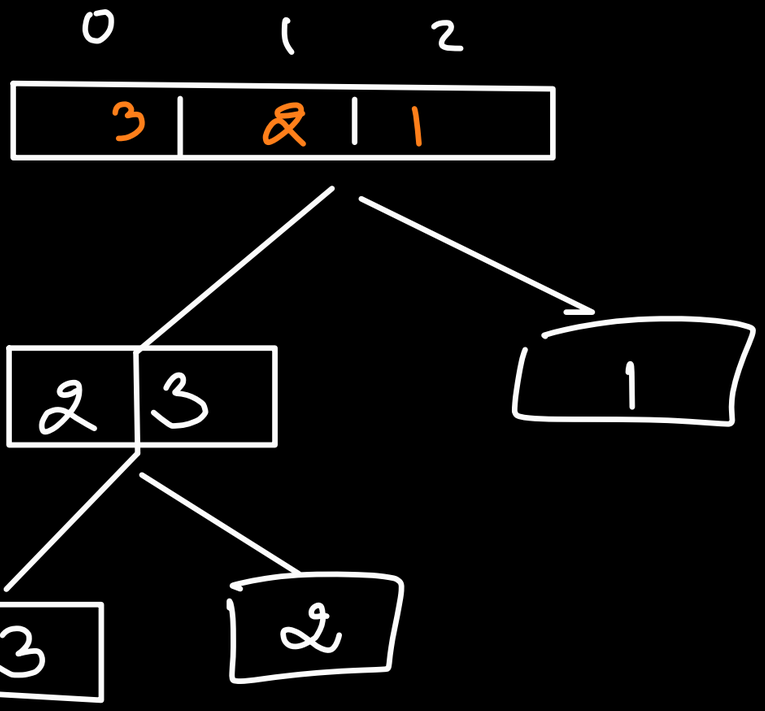
↓

merge ( )

applies merge sort & arrange the elements in inc order $[i, j]$

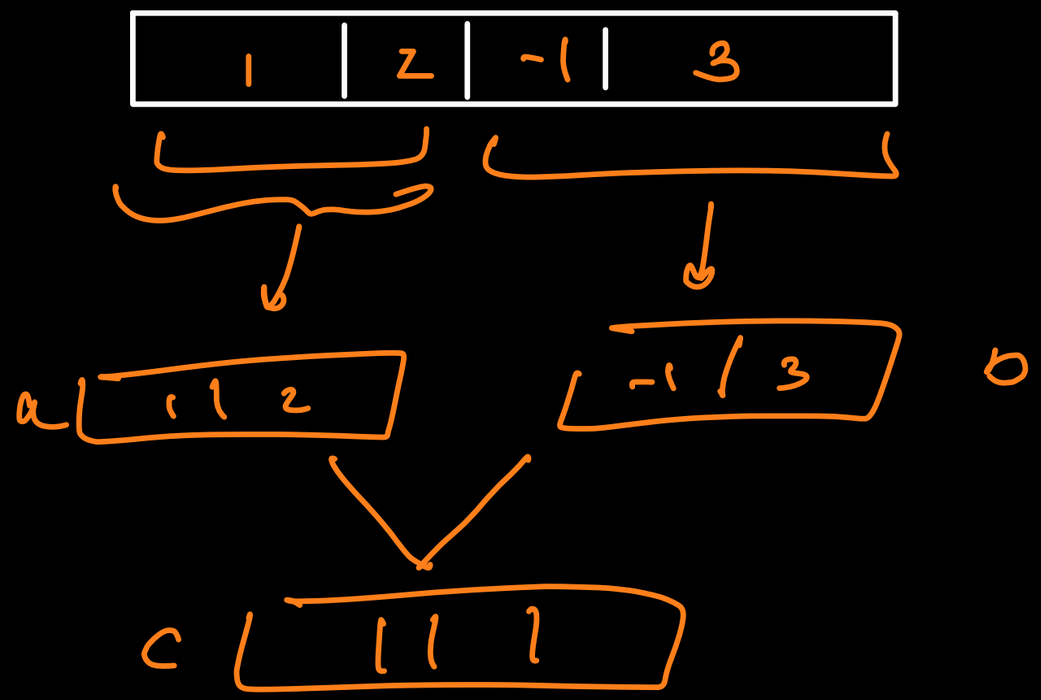
JOIN THE DARKSIDE





$\frac{0+2}{2} \rightarrow 1$

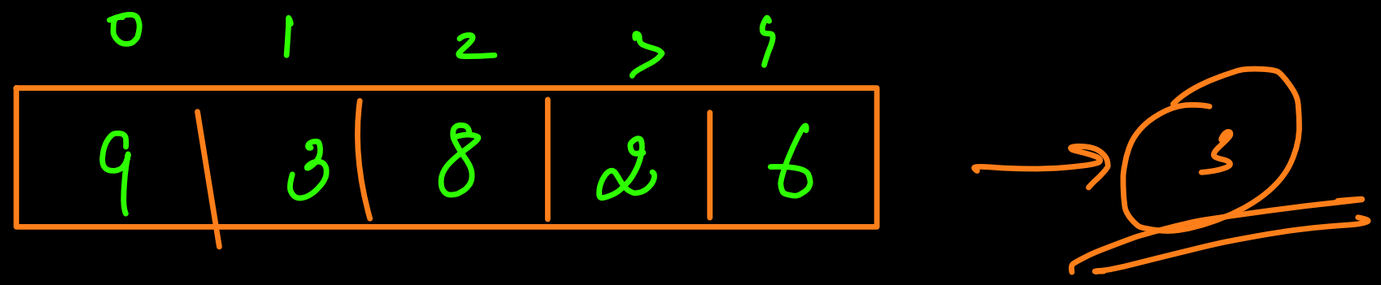
$f(arr, 0, 2)$



→ merge sort is a recursive algorithm. (call stack)

↳ we are manipulating data in external data structure

& hence merge sort is Not Inplace.

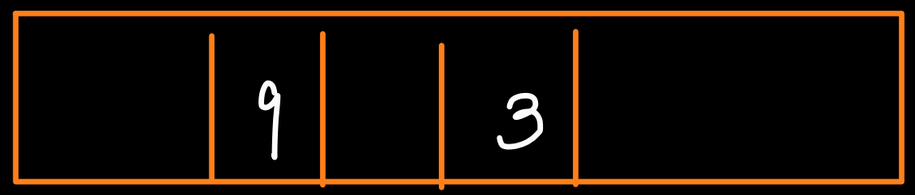


(0,1) $9 > 2 \times 3$

(0,3) $9 > 2 \times 2$

(2,3) $8 > 2 \times 2$

[1,2,3,4] 0



$i < j$

$a_i > 2 \times a_j$

$9 > 2 \times 3$