

Merge Two Sorted

arr1 → 1 3 7 9 11
arr2 → 2 4 5 12 13

final / res array → 1 2 3 4 5 7 9 11 12 13

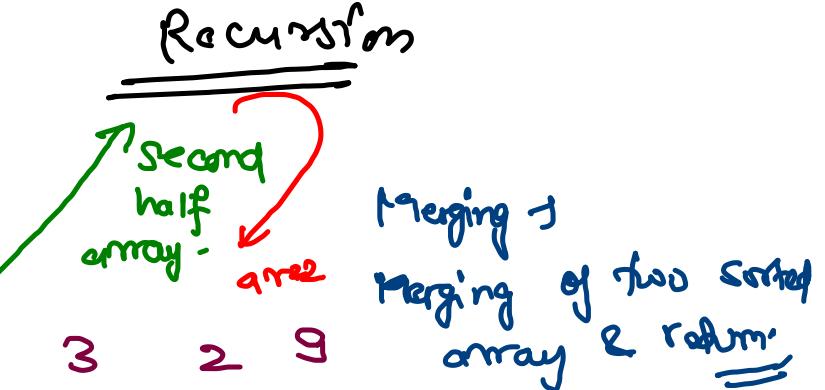
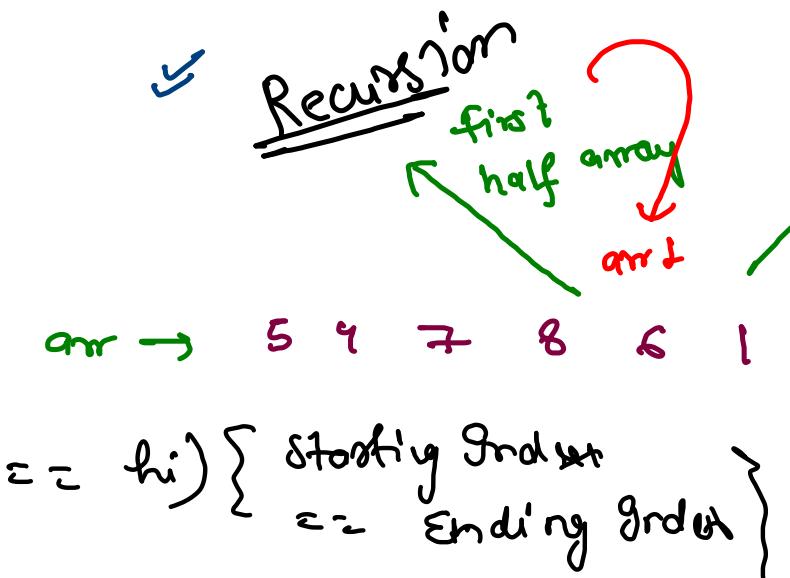
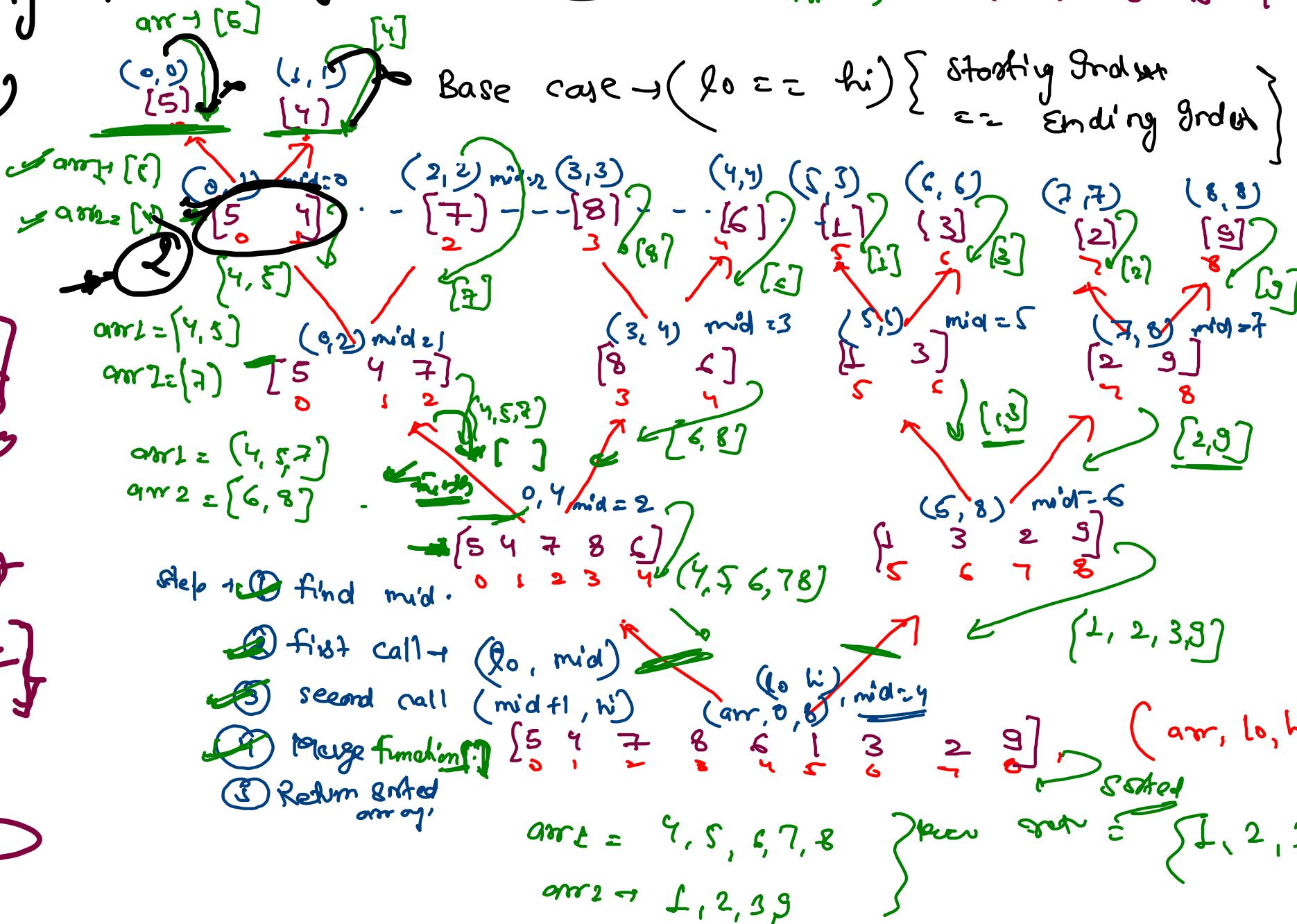
Time Complexity → $O(n \log n)$ proven

Space $O(n)$

in-place sorting ↗
memory requirement ↗

$O(n)$ ↗

ns, re ↗



Garbage collection

PTM ↗

assume if wrong

Mark swap ↗
for i → [faith] (lo, mid)
Ex- ↗

(mid+1, hi) ↗

Merging work is done by mr.

{1, 2, 3, 4, 5, 6, 7, 8, 9}

Merge Two Sorted

arr1 → 1 3 7 9 11
arr2 → 2 4 5 12 13

final / res array → 1 2 3 4 5 7 9 11 12 13

Time Complexity → $O(n \log n)$ proven

Space $O(n)$

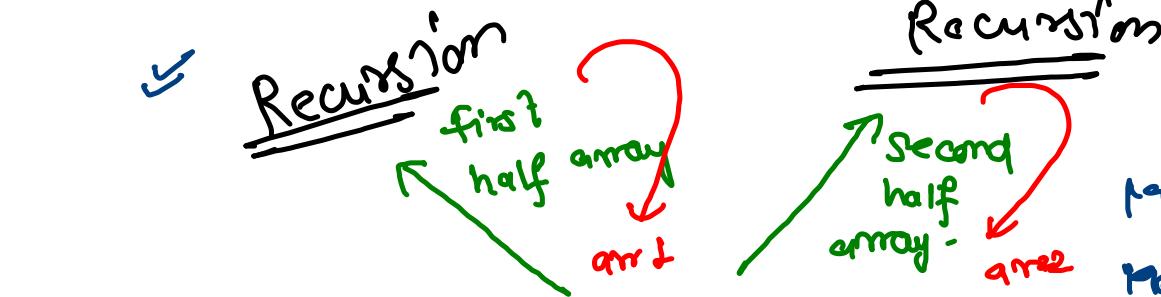
in-place sorting ↗

Memory requirement

$O(n)$

ns, re

→



arr → 5 9 7 8 6 1 3 2 9

Merge ↗
Merging of two sorted array & return

Garbage collection

PTMI

assume if wrong

Mark swap ↗
for i = 1 to n-1

Ex-4

(mid+1, hi) ↗

Merging work is done by mp.

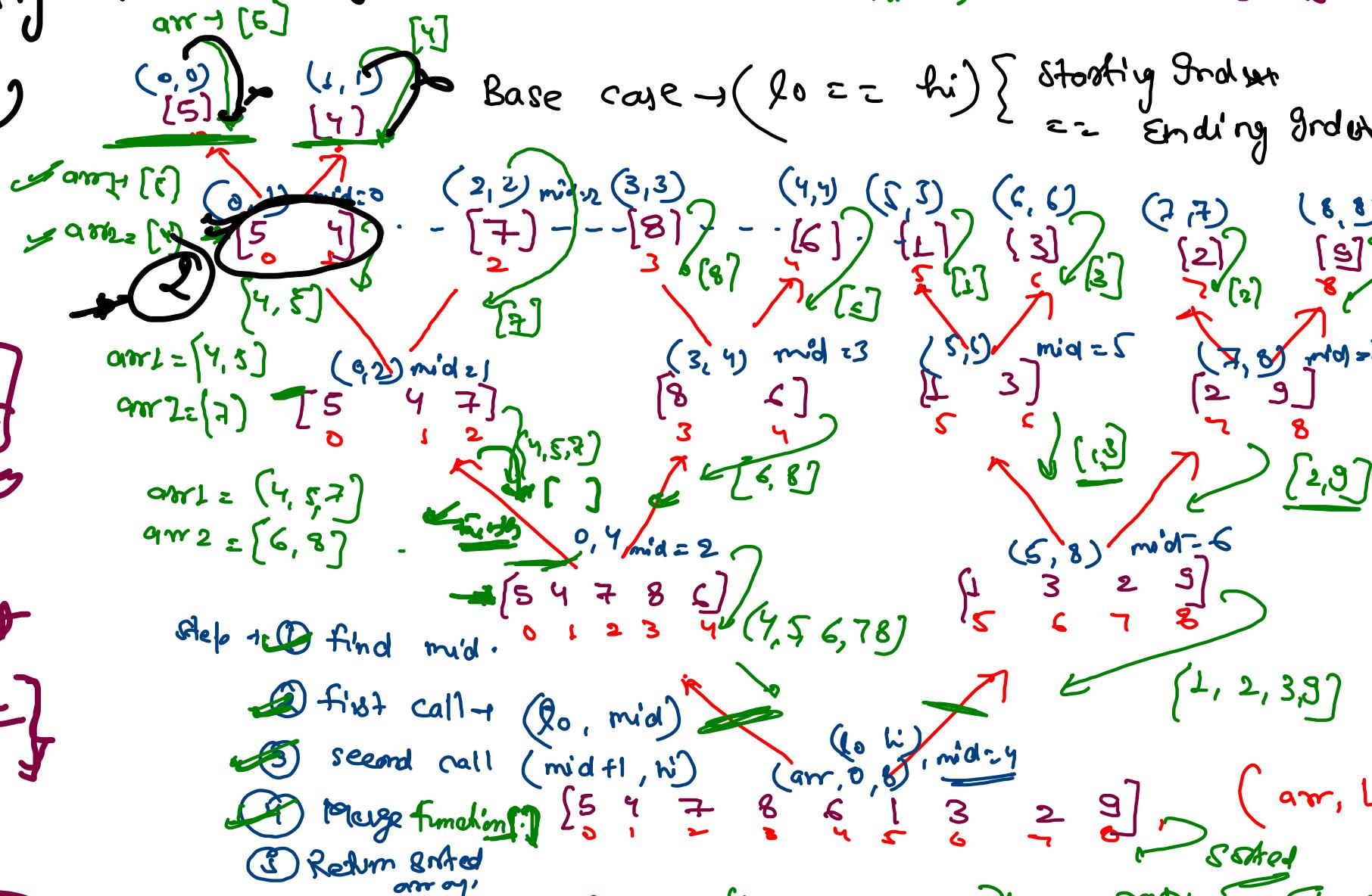
(arr, lo, hi)

sorted ↗

arr1 = 9, 5, 6, 7, 8
arr2 = 1, 2, 3, 9

new arr ↗

{1, 2, 3, 4, 5, 6, 7, 8, 9}

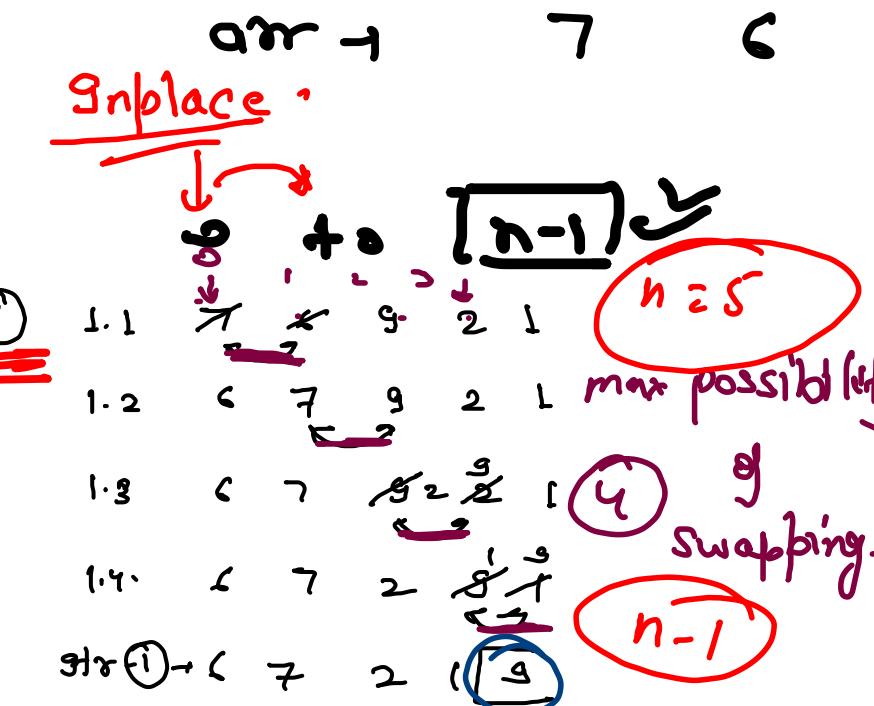


- ① Bubble Sort → Max → move → toward last invalid index
 array is not sorted
 ② Selection Sort
 ③ Insertion Sort → Max is placed at last index (0 to second last)

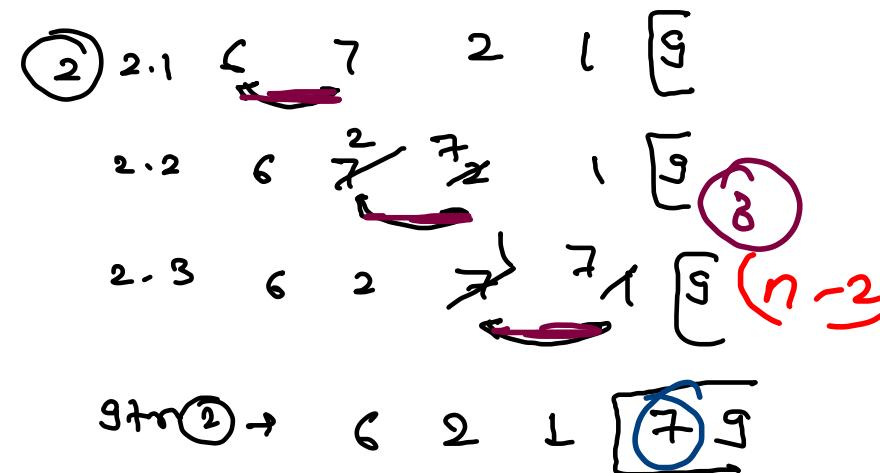
Bubble Sort

Elasayi = 5

gtr = 1
 $j=0$
 $n-gtr$

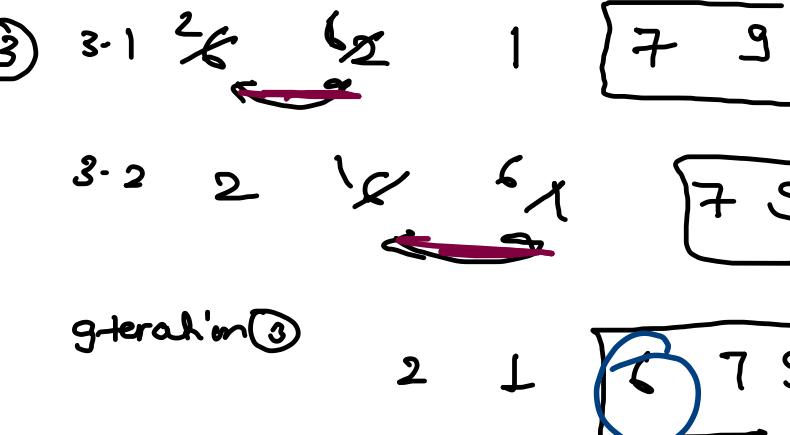


arr.length-1



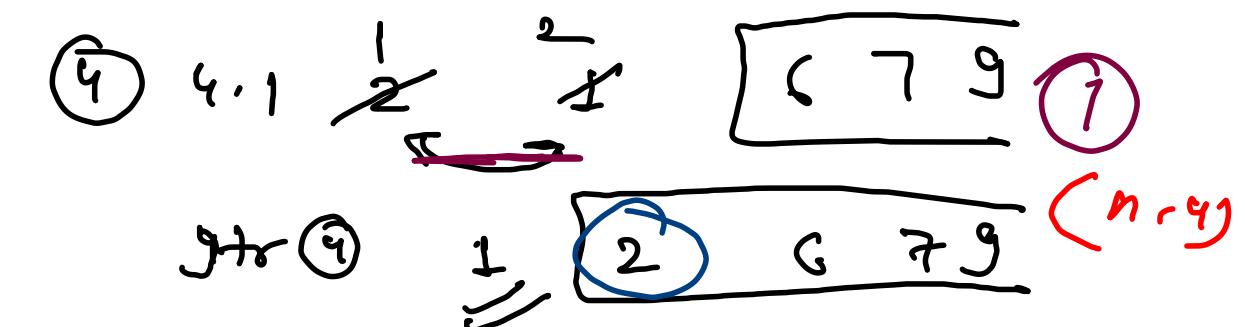
0 1 2 3 4.

$$\text{Time} = 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2} = \frac{n^2 + n}{2}$$



$$(n=3) = O(n^2)$$

Space → O(1)



Electricity issue