* Missing Element? {1, 2, 3, 5, 6, 7} or {1, 2, 3, 4, 5, 6} (C1, C2) → 7

(I) if arr[mid] = mid+1
  ↳ right s = m+1

(II) if arr(mid] != mid+1
  ↳ left
  ↳ checking
    if (mid == 0 || arr[mid-1] == mid)
      ↳ mid+1;

(III) return n+1; (arr.length)

Bubble Sort

(4) [7 | 6 | 4 | 3] (indices 0, 1)

I
6 7 4 | 3
6 4 7 | 3
6 4 3 | 7
US | S

II
6 4 | 3 7
4 6 | 3 7
4 3 | 6 7
US | S

n → (n−1) iterations

III
4 3 6 7
3 4 6 7

1 → n−1 → 4−1 → 3; n−2 (2)

for → iterations
for → swap
f → (n) × (n) = n^2
f

(n^2) Selection Sort (Swap)

[5 | 2 | 9 | 1 | 6 | 4] (n−1)
0 1 2 3 4 5

min index = i
j = i+1 till n

[1 | 2 | 9 | 5 | 6 | 4]

f
1 2 4 5 6 9
f

(n^2) Insertion Sort (shift)

[8 | 2 | 1 | 4 | 3 | 0] (i−1, arr[i]) temp = 2

[2] | [8 | 1 | 4 | 3 | 0]
Sorted | unsorted

[ ] 2 8 4 3 0

UNO

O(n)
n−1
shift
O(n)
O(n^2)

* Merge two sorted Arrays : Two Pointer Approach

i = 0
j = 0
k = 0

a1 = [1 | 3 | 5 | 7 | 9] (0 1 2 3 4)
a2 = [2 | 4 | 6] (0 1 2)

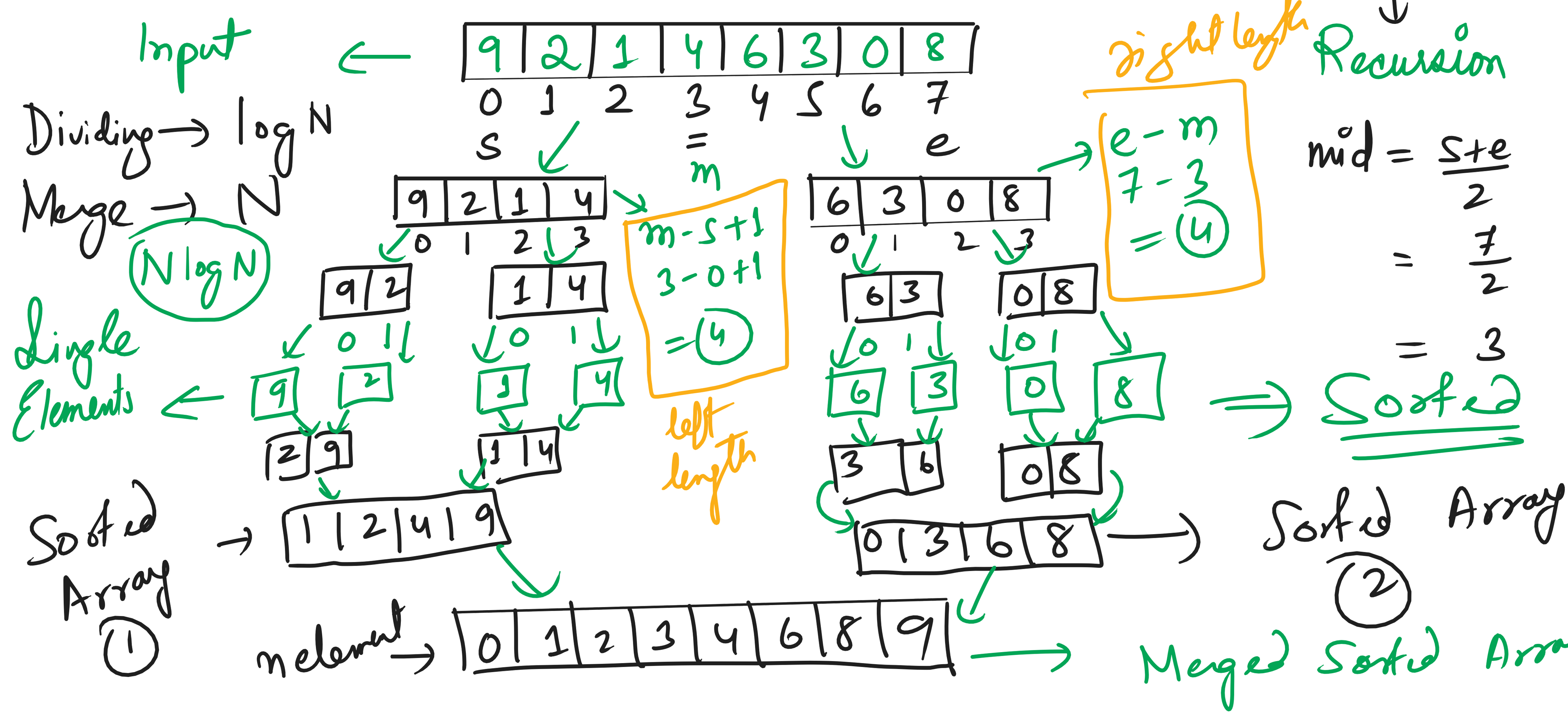
if (a1[i] < a2[j])
  a3[k] = a1[i]
  k++, i++
else (a2[j] < a1[i])
  a3[k] = a2[j]
  k++, j++

while (i < n1)
  arr[k++] = a1[i++]
while (j < n2)
  same;

a3 = [1 | 2 | 3 | 4 | 5 | 6 | 7 | 9]
0 1 2 3 4 5 6 7
k k k

Merge Sort Algorithm :→

Divide & Conquer → Repetitive Division ↓ Recursion



mid = (s+e)/2
= 7/2
= 3

⇒ Sorted

* OOPs + Exception Handling + File Handling

Spring MVC | Boot
FE + SQL   MySQL

↳ CRUD
create Folder
       File
delete Folder
       File
text ⊂ read File
excel   write File