

Divide & conquer

- 1. Divide the bigger problems into smaller subproblems
- 2. conquer / solve those subproblems via Recursion
- 3. combine the solutions of subproblems to get



Optional

function name starting index
ending index

T(n) ————— DAC (arr, i, j) ↴

if (small(arr, i, j)) ↴

Logical understanding

C ————— return solution (arr, i, j)

↳

else ↴

1. Divide

C ————— m = Divide (arr, i, j)

Recursion ↴

2. Conquer

Recursion ↴

— T(n/4) ————— b = DAC (arr, i, m)

— T(n/4) ————— c = DAC (arr, m+1, j)

3. combine

↳ C

return (combine (b, c))

50	20	10	27	15	35
0	1	2	3	4	5

$$\begin{cases} i=0 \\ j=5 \end{cases}$$

Sort

Small Problem

→ single element



$i=0$ }
 $j=0$ } $i==j$
 ↓
 → single element

Small Problem

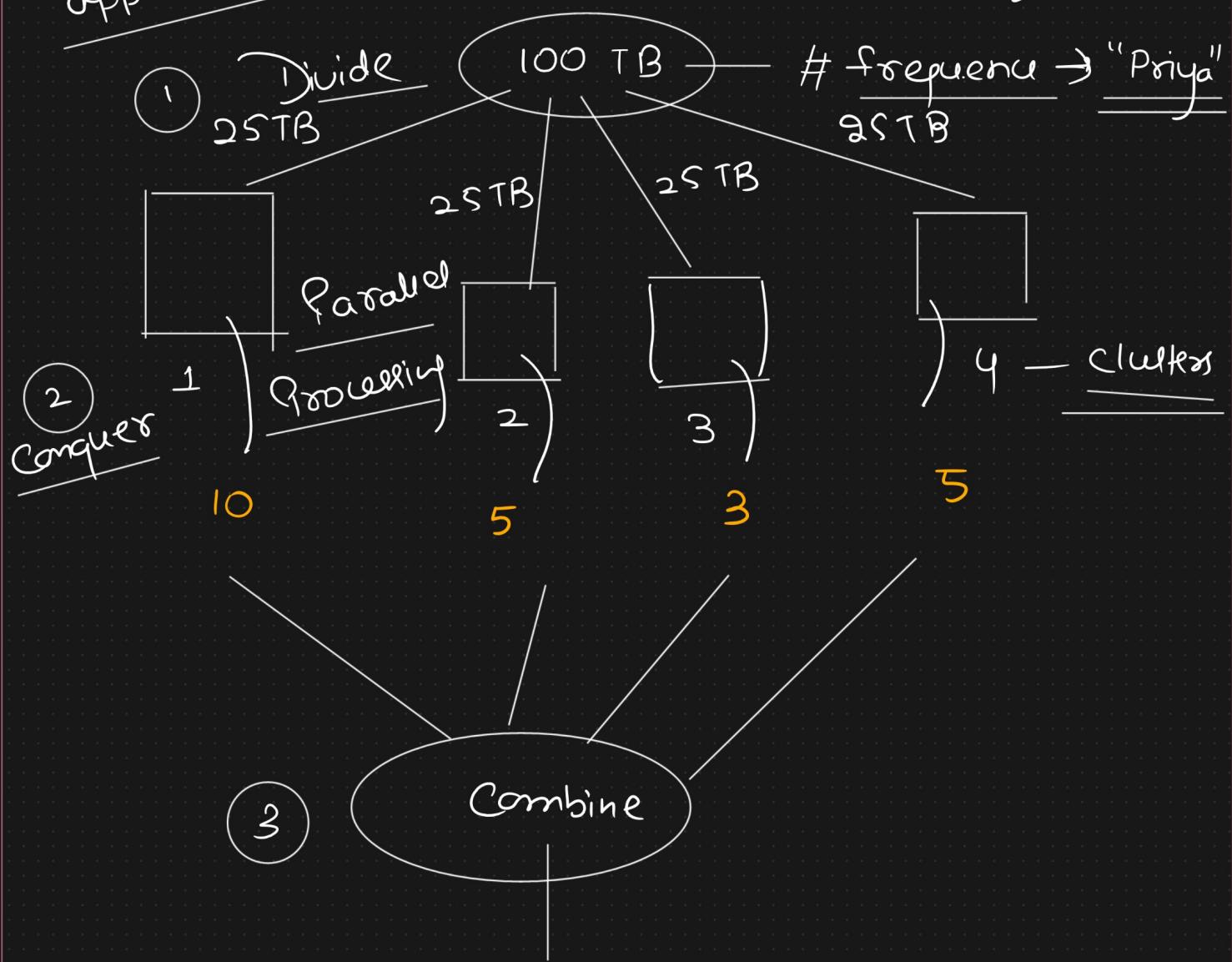
Recurrence Relation

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

Real time
application

Big Data \rightarrow Divide & conquer

\hookrightarrow Data \rightarrow too huge

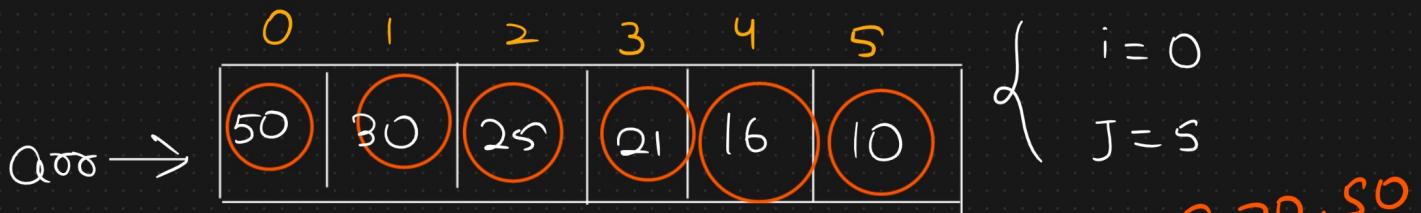


$$10 + 5 + 3 + 5$$

$$= 23$$

MergeSort

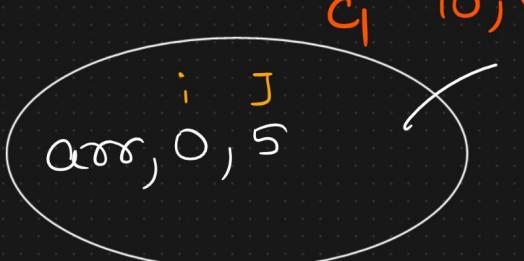
$n = 6$



Recursive tree

$$\text{mid} = \frac{0+5}{2}$$

$$= 2$$



c_1 10, 16, 21, 25, 30, 50
ROOK ↓
Node

sorted array

$$\begin{aligned} &2 - 0 + 1 \\ &= 3 \end{aligned}$$

$$arr, i, \underline{\text{mid}}$$

$$arr, \underline{\text{mid}+1}, j$$

20, 50 c_3

arr, 0, 1

c_4

arr, 0, 0

50

arr, 1, 1

c_5

Leaf

Node

16, 21 c_8

c_9

arr, 3, 3

21

arr, 4, 4
10
16

c_{11}

arr, 5, 5

c_6

arr, 2, 2

c_6

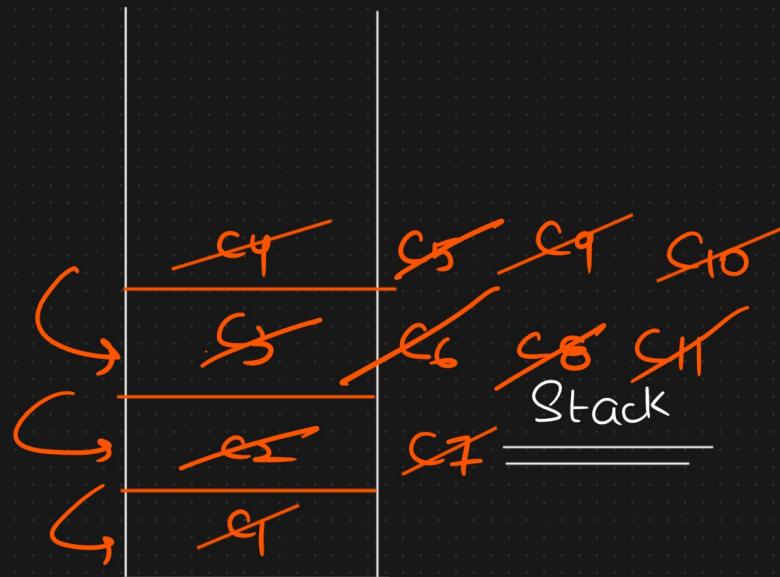
c_5

25

Leaf

Node

Recursive function call



~~mergeProcedure~~ → combine part in
give us a sorted array

MergeSort
 $T(N)$ (function name)
mergeSort(arr, i, j) if ($i < j$) Big Problem

1) Divide

C _____ $\text{mid} = i + (j - i) // 2$

2) conquer

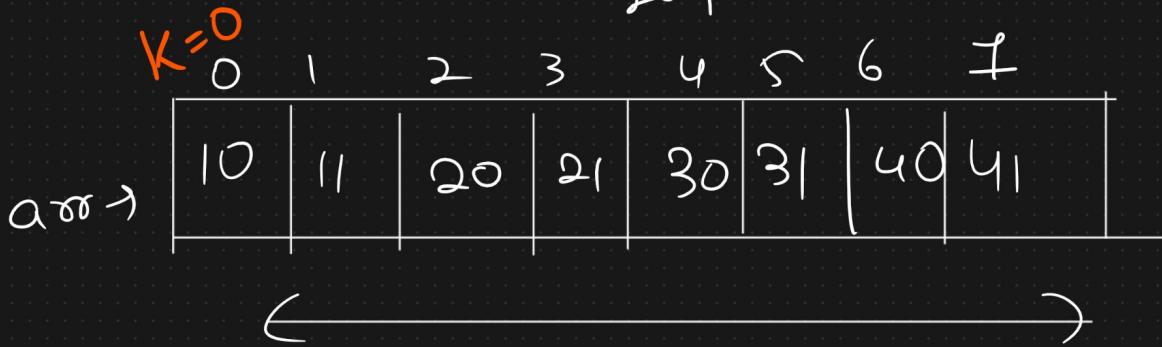
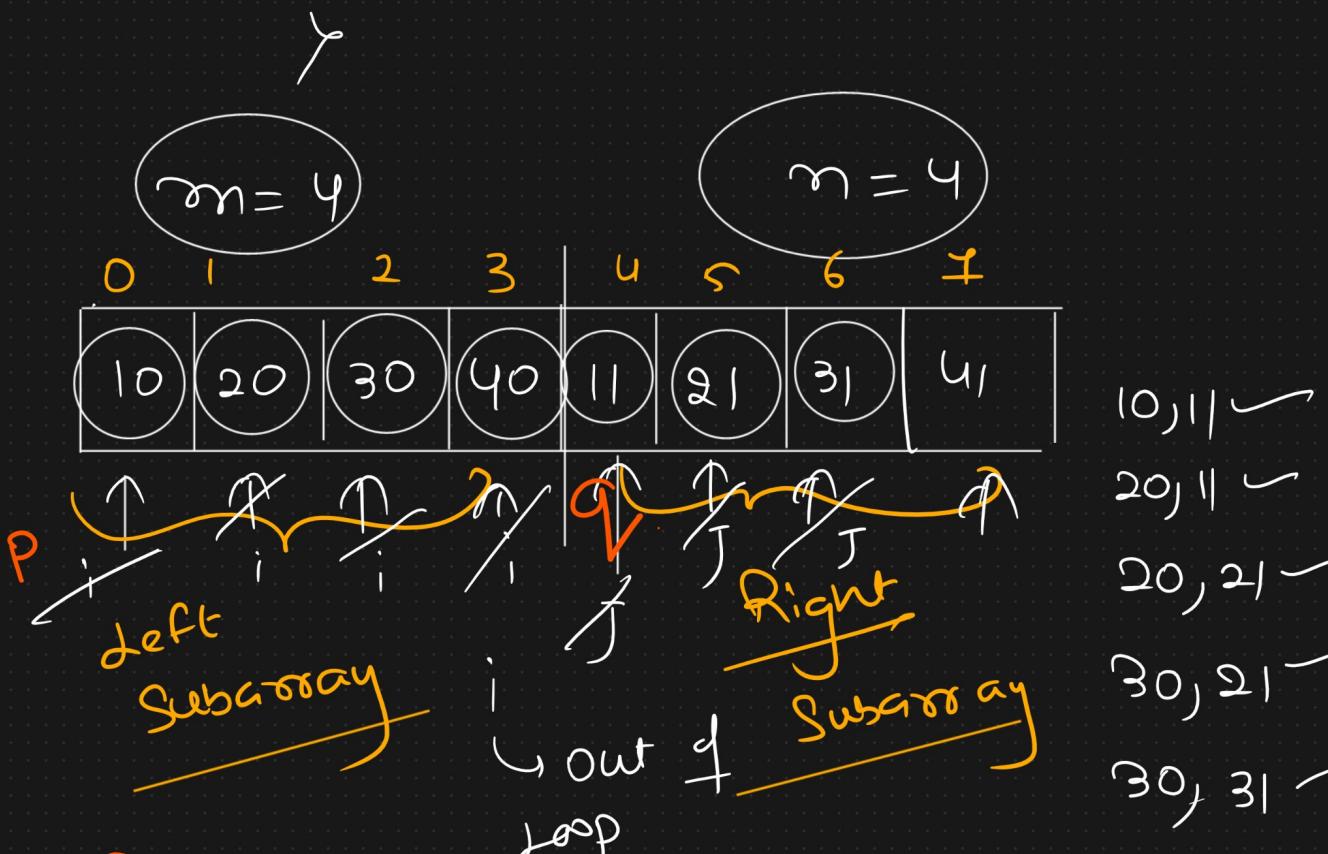
Left $T(N/2)$ mergeSort(arr, i, mid)

Subtree $T(N/2)$

Right
Subtree

$O(N)$ ————— mergeProcedure(arr, i, mid, j)

3) Combine



Worst Case \rightarrow # comparisons

$$m+n-1$$

$$m=4 \quad n=2$$

$$4+4-1 = 7$$

$0 \quad 1 \quad 2 \quad 3$	$4 \quad 5 \quad 6$	$\{10, 5\}$
$10 \quad 20 \quad 30 \quad 40$	$5 \quad 6$	$\{10, 6\}$

Best case \rightarrow # comparison

$$\min(m, n)$$

$$\min(4, 2) = 2$$

$0 \quad 1 \quad 2 \quad 3$	$4 \quad 5$	$\min(m, n)$
$5 \quad 6 \quad 10 \quad 20 \quad 30 \quad 40$		$\min(4, 2) = 2$

Time Complexity

\rightarrow # comparison + # move

$\rightarrow O(m+n)$

$\rightarrow O(N)$

$$N = m+n$$

Recurrence Relation

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

(combine
(merge) result)

| Size of
Subproblems

$$= \underline{\underline{\mathcal{O}(N \log N)}}$$

Space complexity = $\mathcal{O}(N)$

↳ Extra Space

Output^c sorting

algo