If the recurrence is of the form $\boxed{T(n) = aT(\frac{n}{b}) + \Theta(n^k \log^p n)}$, where $a \geq 1, b > 1, k \geq 0$ and p is a real number, then:

1) If $a > b^k$, then $T(n) = \Theta(n^{\log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \Theta(n^{\log_b^a} \log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \Theta(n^{\log_b^a} \log\log n)$
   c. If $p < -1$, then $T(n) = \Theta(n^{\log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \Theta(n^k \log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

Master theorem –

$\log^p n \{ \log^0 n) = 1$

Recurrence Relation

↓ Master thm

Time Complexity –

$\boxed{a^0 = 1}$

① $T(n) = \overbrace{3T(n/2)}^{} + n^2$ , $aT(n/b)$   $\Theta(n^k \log^p n)$ → formulas

$a = 3$, $b = 2$, $k = 2$, $p = ??$ $p = 0$   $n^2$ → question

$a < b^k$ | 3rd condition → a wala condn.

$3 < 2^2 = 4$

$T(n) = \Theta(n^k \log^p n) = \Theta(n^2 . \log^0 n) = \Theta(n^2 . 1)$

$= \Theta(n^2)$

② $T(n) = 4T(n/2) + n^2.$   $aT(n/b)$   $\Theta(n^k \log^p n)$

$n^2$

$a = 4$, $b = 2$, $k = 2$, $p = 0$

$a = b^k$

$4 = 2^2$

Condn 2 → condn (a) $T(n) = \Theta(n^{\log_b^a} \log^{p+1} n)$

$= \Theta(n^{\log_2^4} . \log^{0+1} n)$

$= \Theta(n^2 . \log^1 n)$

$\log_2 4$

$= \log_2 2^2$

$= 2 \log_2 2$

$= 2.$

$n^2 \log n$

## Divide & Conquer –

Technique to design algorithms.

$\boxed{\text{Big Problem}}$

(... sub problems)

Eg –

P: $1 + 2 + 3 + 4 = 10$

$P_1$ $P_2$ . . . . . $P_k$ (Small sub-problems)

$S_1$ $S_2$ $S_k$ (Solution of small sub problems)

Big Solution

$P$: $1+2+3+4 = 10$

$P_1$: $(1+2)$ $P_2$: $(3+4)$

$S_1$: 3 $S_2$: 7

$S$: 10

## Merge Sort - Divide & Conquer Technique.

→ | 6 | 4 | 2 | 1 | 9 | 8 | 3 | 5 |

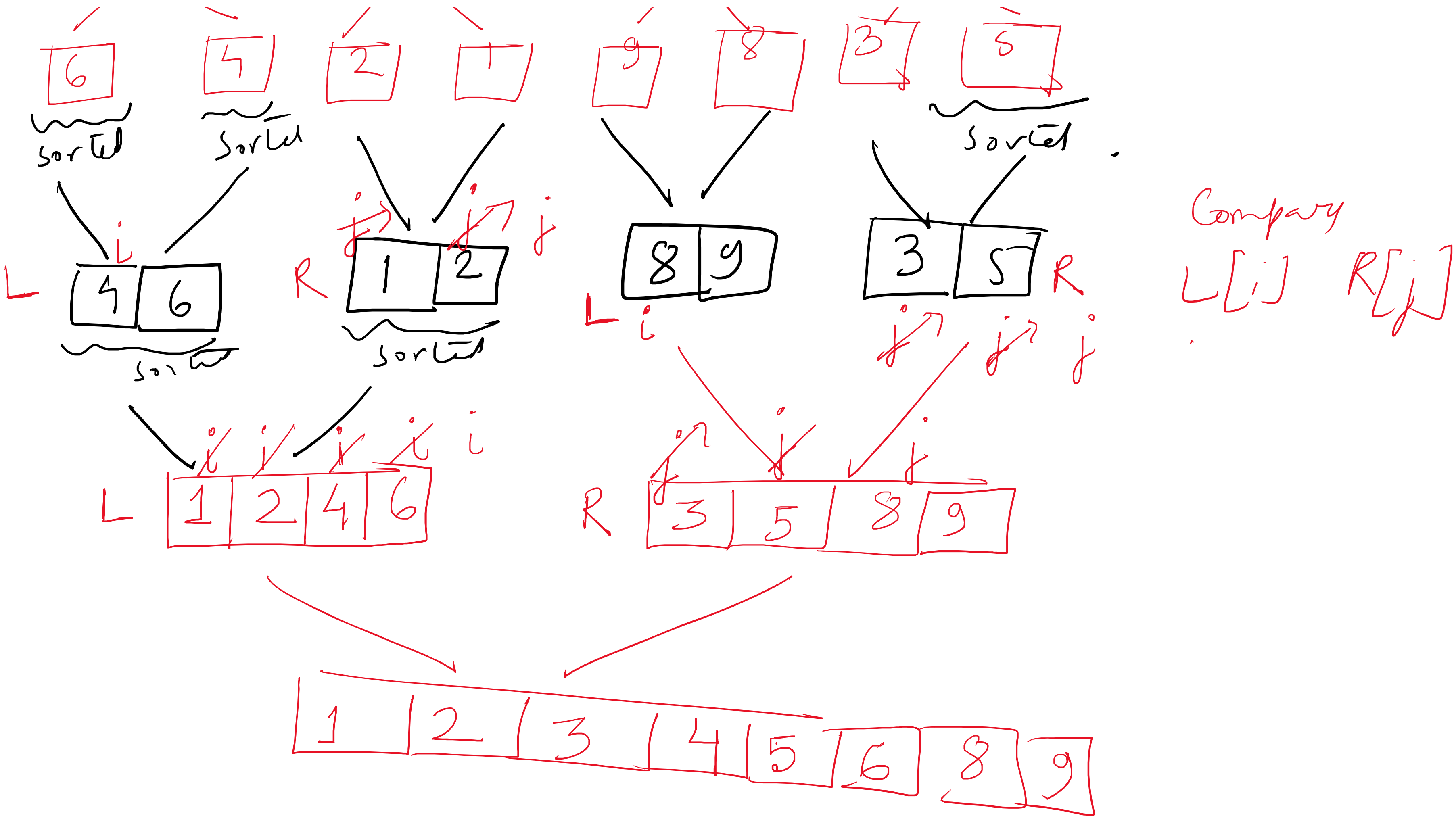
6 4 2 1 9 8 3 5

6 4 2 1 9 8 3 5

| 6 | | 4 | | 2 | | 1 | | 9 | | 8 | | 3 | | 5 |

Note -

1. Repeatedly divide the array into 2 equal parts.
2. Combine/Merge 2 sorted arrays into 1 sorted array.


sorted
sorted
sorted
sorted
sorted
L
R
Compare
L[i] R[j]