If the recurrence is of the form $T(n) = aT(\frac{n}{b}) + \Theta(n^k log^p n)$, 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 $T(n) = \Theta(n^{log_b^a} loglogn)$
   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 $T(n) = O(n^k)$

$T(n) = 3T(n/2) + n^2$

$aT(n/b)$ $3T(n/2)$

$a=3, b=2$

$= \Theta(n^k \log^p n)$ | n^2 O

$k = 2$

$a^0 = 1$

1st step -

a, b, p, k

2nd step

$a < b^k \Rightarrow$

$3 < 4$

Cond^n – 3rd

$p = 1$, $n^2 \log^1 n$

$p = 0$

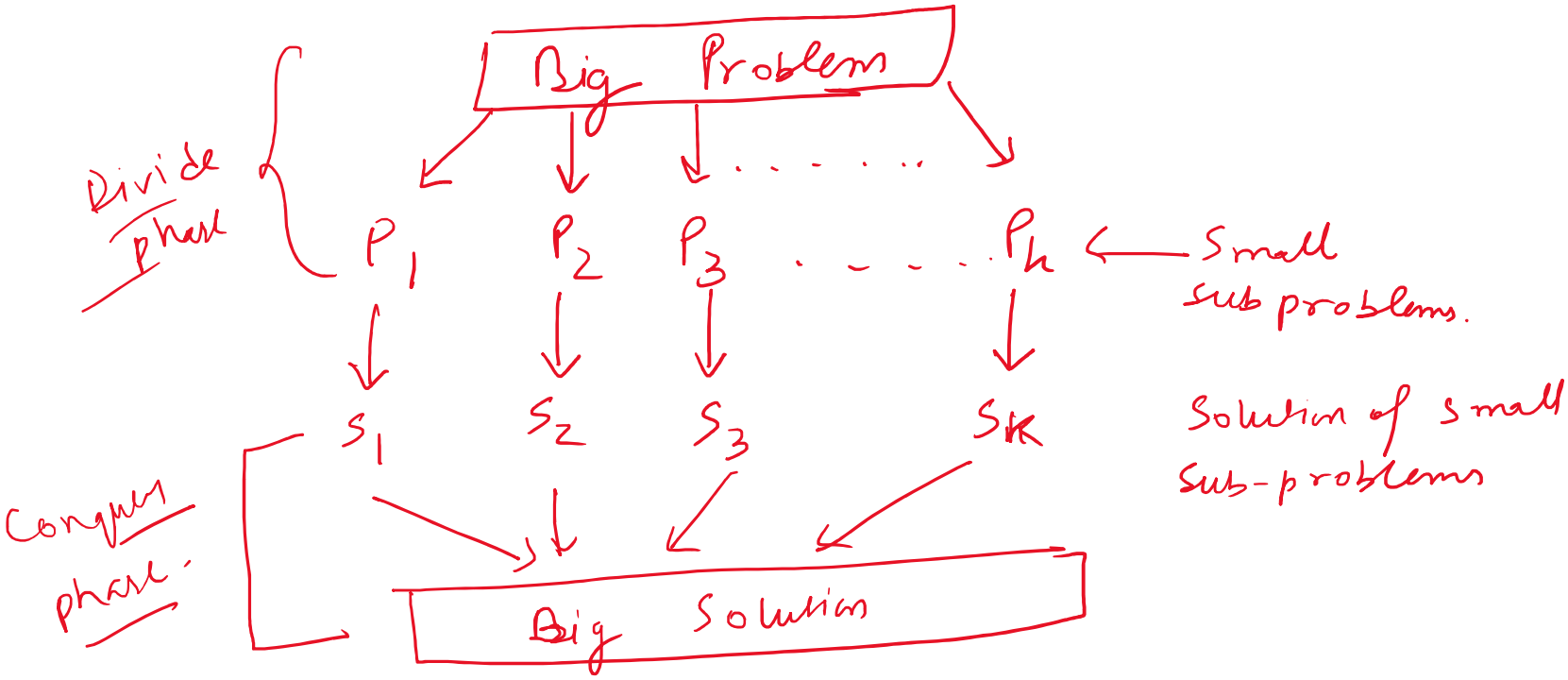
$T(n) = \Theta(n^k \log^p n)$

$= \Theta(n^2 \log^0 n) = \Theta(n^2)$.

$p = 0$,

Cond^n – 3a

Divide & Conquer – Technique to design algorithm.

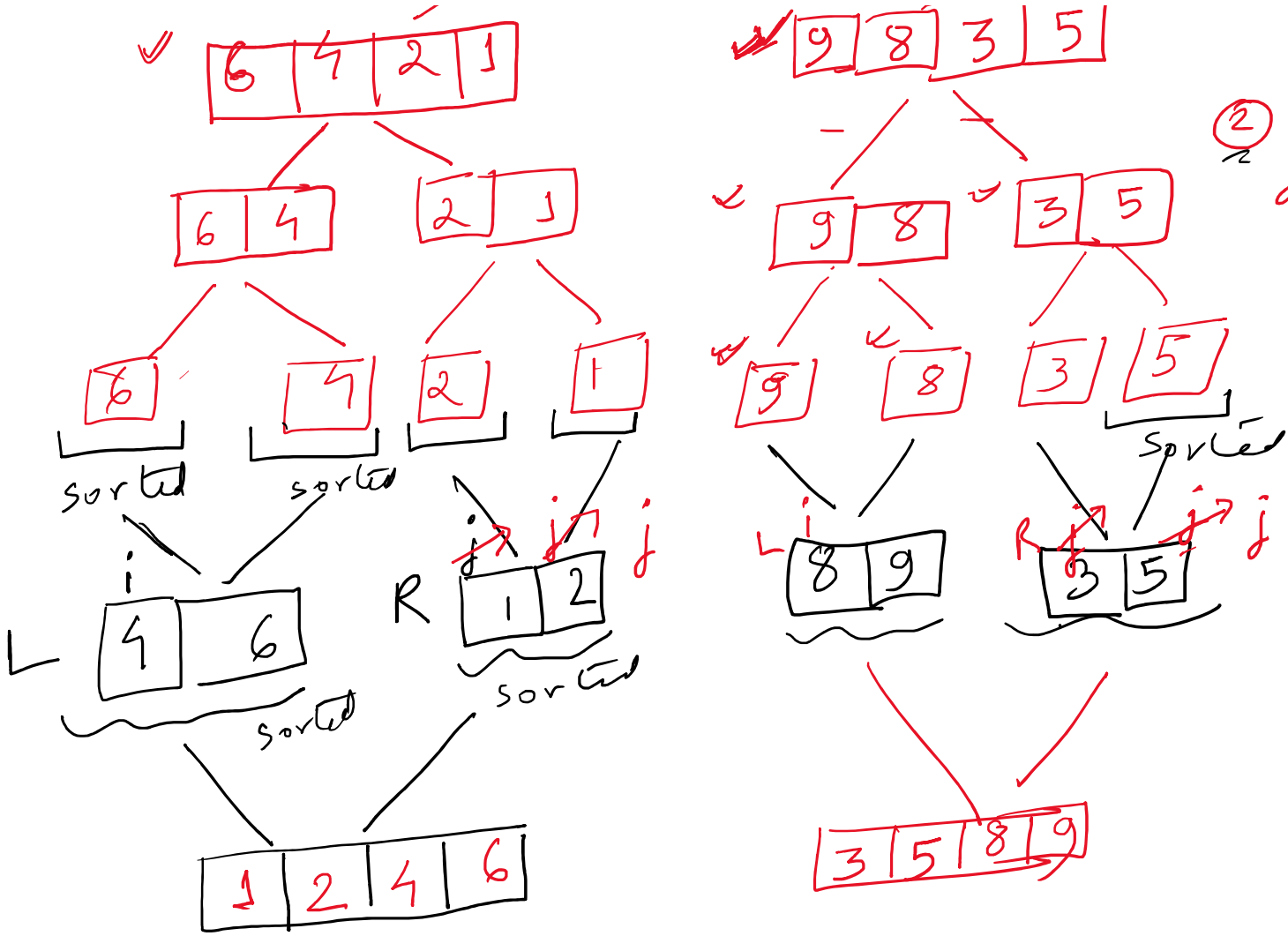


Merge Sort – Sorting algo based on divide & conquer technique.

6 4 2 1 9 8 3 5

Note –

① Repeatedly divide the array into 2 equal parts.



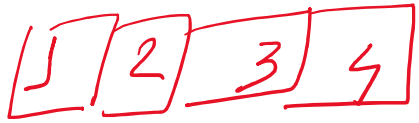
partn.

② Merge 2 sorted arrays into 1 sorted array.

Compare

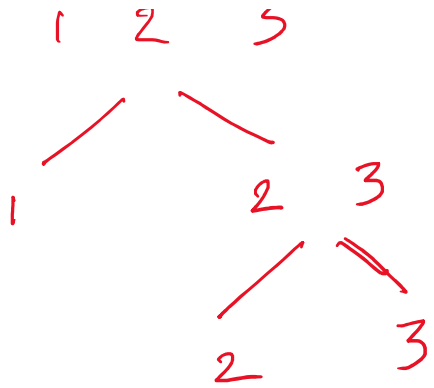
L[i] R[j]

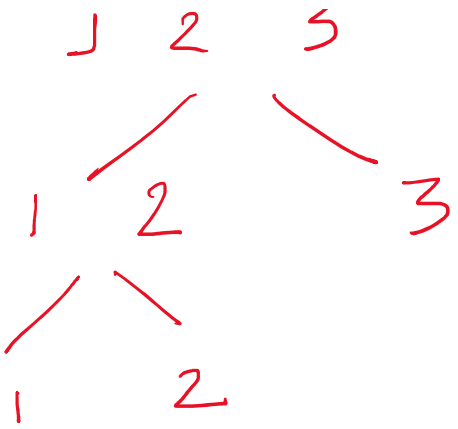
Even no. of elements

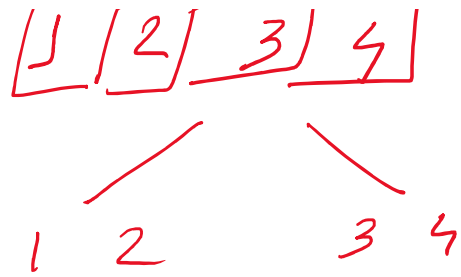


1 2 3

1 2 3


1 2 3
1
2 3
2
3


1 2 3
1 2
3
1
2


1 2 3 4
1 2
3 4