Master Theorem -

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 $T(n) = O(n^k)$

$T(n) = T(n-1) + 2$

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

$\boxed{aT(n/b)}$ $3T(n/2)$

$n^k \log^p n$

$a=3, b=2, k=2, p=0$

$a=3, b=2, k=2, b^k = 2^2 = 4$

$a < b^k$ Cond^n 3rd.

$p = 0$, Cond^n - 3a.

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

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

$T(n) = 4T(n/2) + n^2$

① $a=4, b=2, k=2, p=0$

② $a=4, b^k = 2^2 = 4, a = b^k$

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

② $a=4, b^k = 2^2 = 4$, $a = b^k$

Condⁿ- 2nd.

$p = 0 > -1$ (2 a)

2) If a= b^k

a. If $p > -1$, then $T(n) = \Theta\left(n^{\log_b^a} \log^{p+1} n\right)$

b. If $p = -1$, then $T(n) = \Theta\left(n^{\log_b^a} \log\log n\right)$

c. If $p < -1$, then $T(n) = \Theta\left(n^{\log_b^a}\right)$

3) If $a < b^k$

a. If $p \geq 0$, then $T(n) = \Theta(n^k \log^p n)$

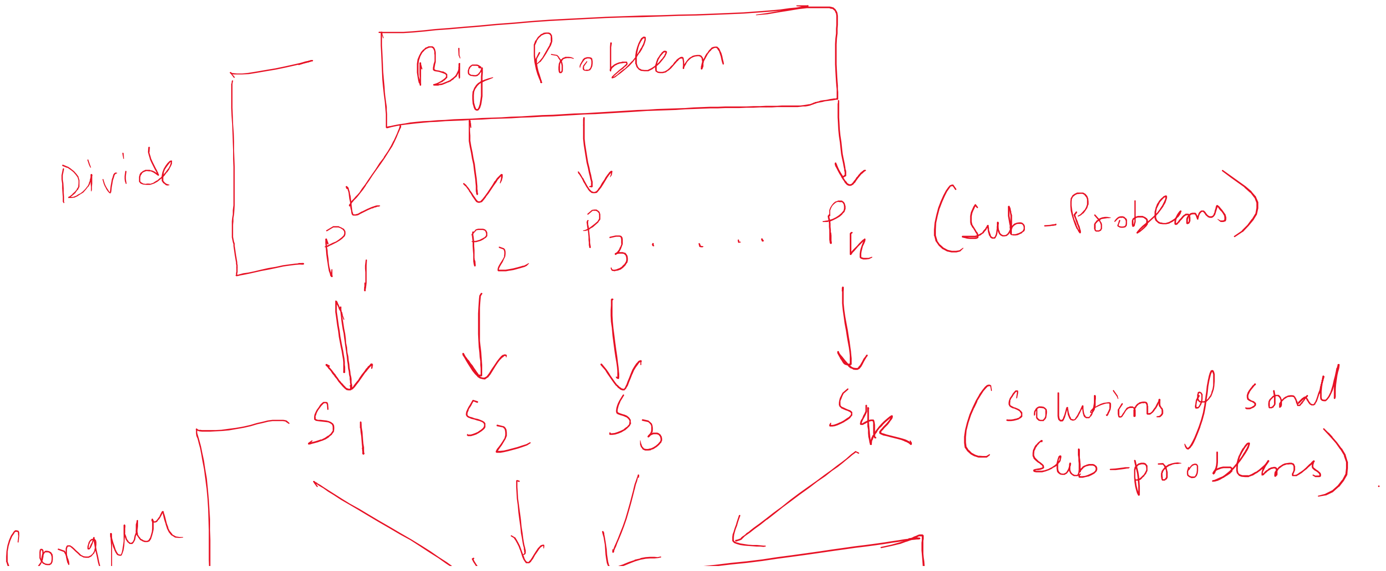
b. If $p < 0$, then $T(n) = O(n^k)$

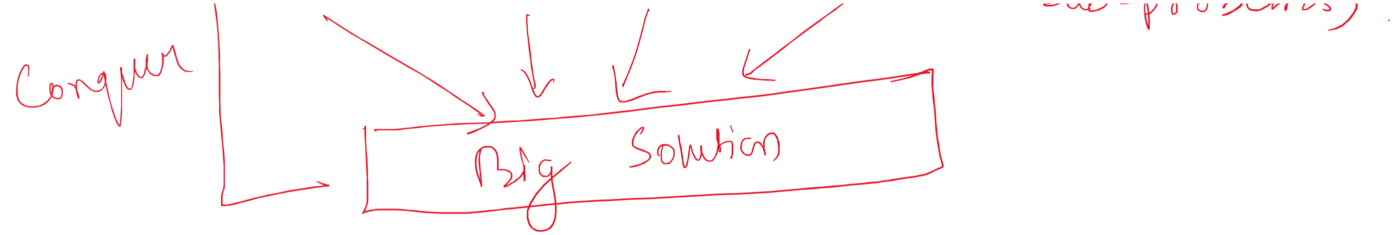
$$T(n) = \Theta\left(n^{\log_b a} \cdot \log^{p+1} n\right) = \Theta\left(n^{\log_2 4} \cdot \log^{0+1} n\right) = \Theta$$

$$\log_2 4 = \log_2 2^2 = 2 \times \log_2 2 = 2$$

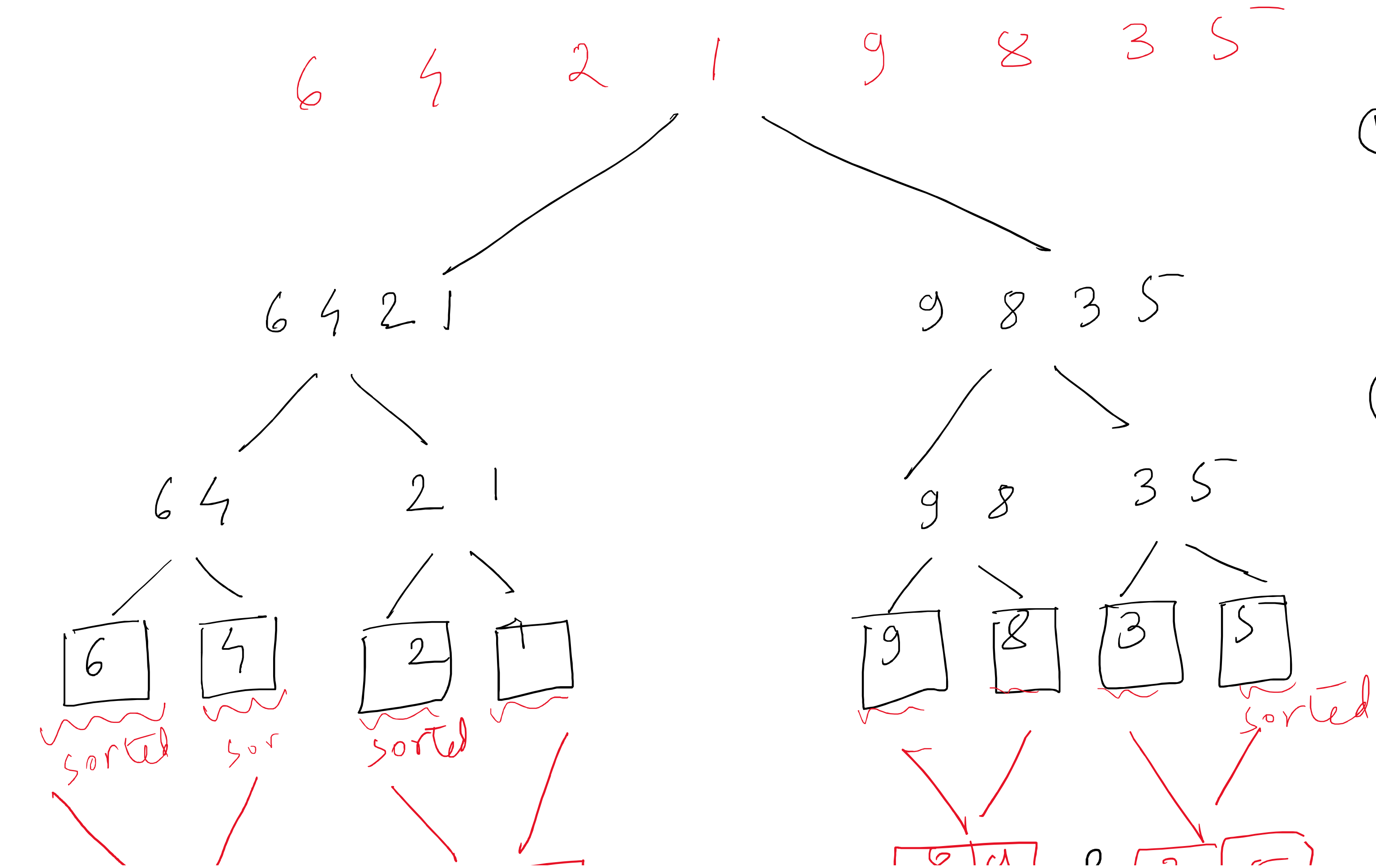
$$= \Theta\left(n^2 \cdot \log n\right)$$

Divide & Conquer - Technique to ~~solve~~ design algorithms.





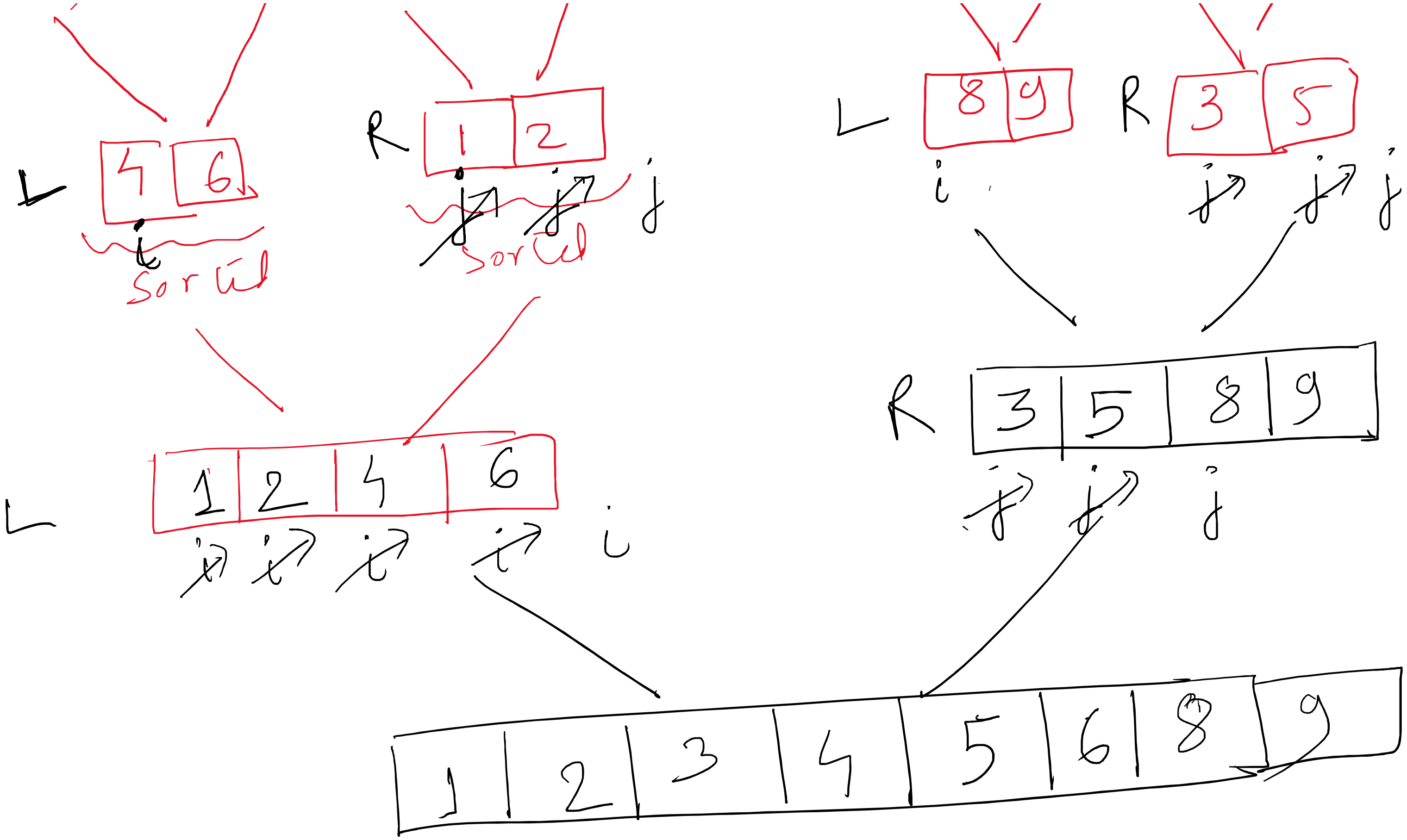
Merge Sort — Based on divide & conquer.



(1) Repeatedly divide the array into 2 equal parts.

(2) Combine 2 sorted arrays into 1 sorted array.

Compare



Compare

L[i] R[j]