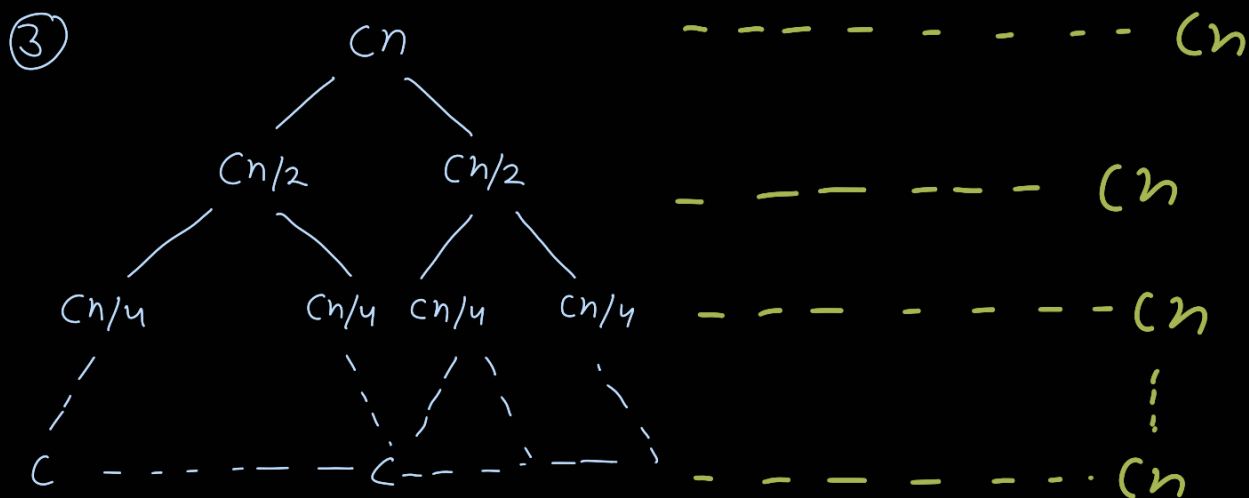
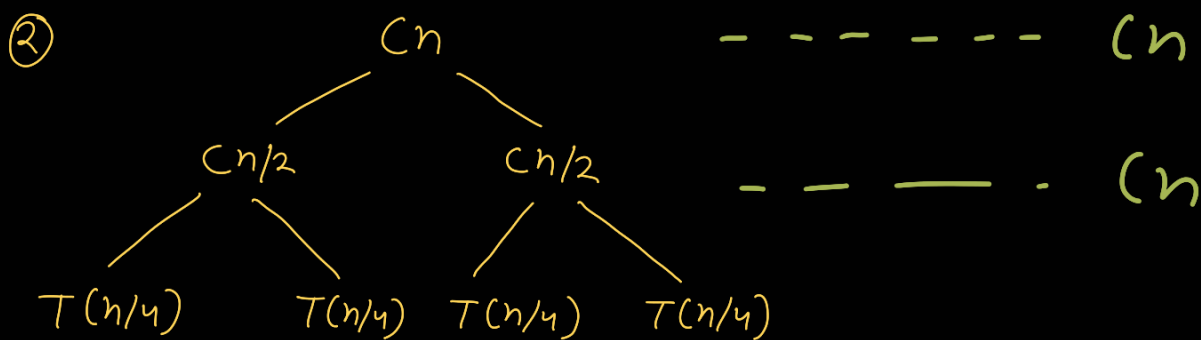
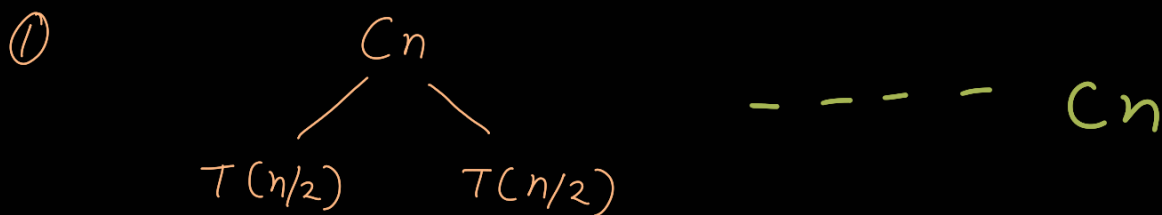


- We consider the recursion tree and compute the total work done.
- We write non-recursive part as root of the tree and write the recursive part as children.
- We keep expanding until we see a pattern.

$$T(n) = 2T(n/2) + Cn$$

$$T(1) = C$$



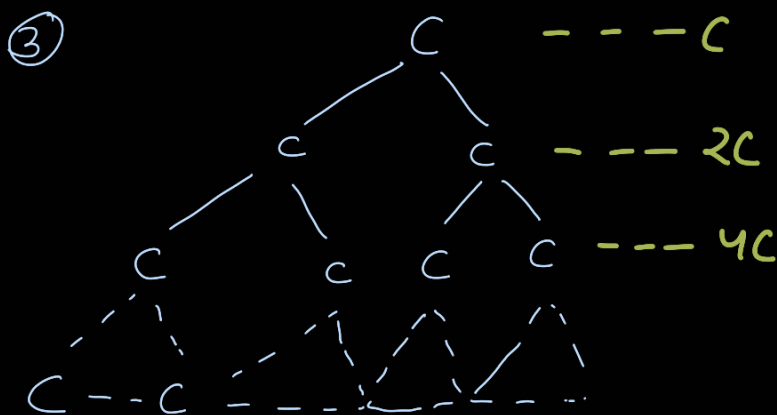
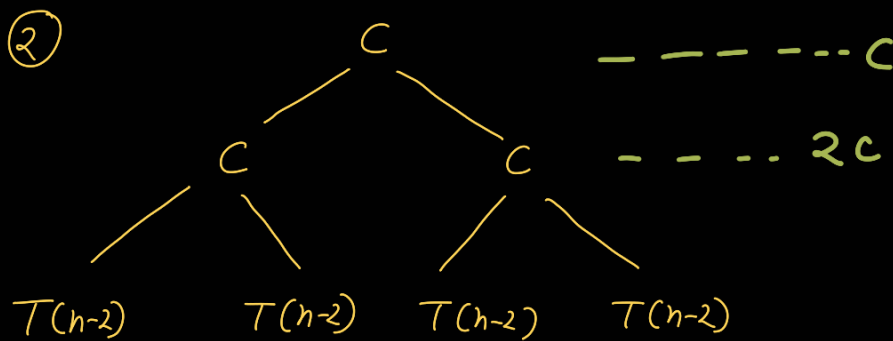
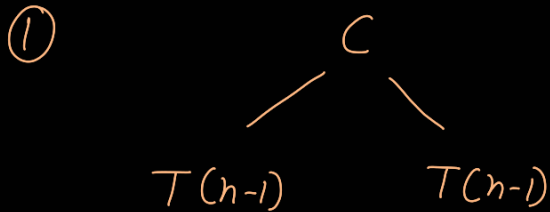
$$\underbrace{Cn + Cn + Cn + \dots + Cn}_{\Theta(\log n)}$$

$$= \Theta(n \log n)$$

More Example Recurrences

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

$$T(1) = C$$



$$C + 2C + 4C + \dots$$

$$C[1 + 2 + 4 + \dots]$$

$$C \left[\frac{1 \times (2^n - 1)}{2 - 1} \right]$$

$$\Rightarrow \Theta(2^n)$$

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

$$T(1) = C$$

①

$$\begin{array}{c} C \\ | \\ T(n/2) \end{array}$$

②

$$\begin{array}{c} C \quad \text{-----} C \\ | \\ C \quad \text{-----} C \\ | \\ T(n/4) \end{array}$$

③

$$\begin{array}{c} C \quad \text{-----} C \\ | \\ C \quad \text{-----} C \\ \vdots \\ C \quad \text{-----} C \end{array}$$

$$\underbrace{C + C + \dots + C}_{(\log_2 n) + 1}$$

$$= \Theta(\log_2 n)$$

$$T(n) = 2T(n/2) + C$$

$$T(1) = C$$

①

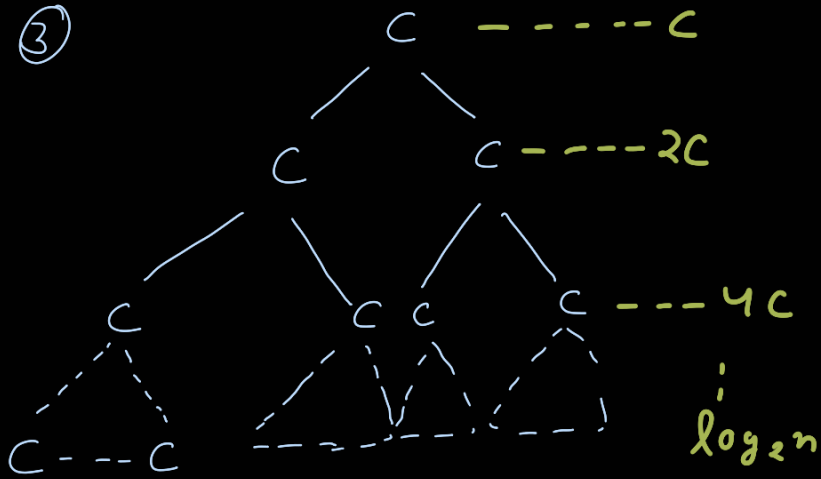
$$\begin{array}{c} C \quad \text{-----} C \\ / \quad \backslash \\ T(n/2) \quad T(n/2) \end{array}$$

②

$$\begin{array}{c} C \quad \text{-----} C \\ / \quad \backslash \\ C \quad \quad C \quad \text{-----} 2C \\ / \quad \backslash \quad / \quad \backslash \end{array}$$

$$T(n/4) \quad T(n/4) \quad T(n/4) \quad T(n/4)$$

③



$$C + 2C + 4C + \dots$$

$$\Theta(\log_2 n)$$

$$\Theta\left(\frac{2^{\log_2 n} - 1}{2 - 1}\right)$$

$$= \Theta(n)$$