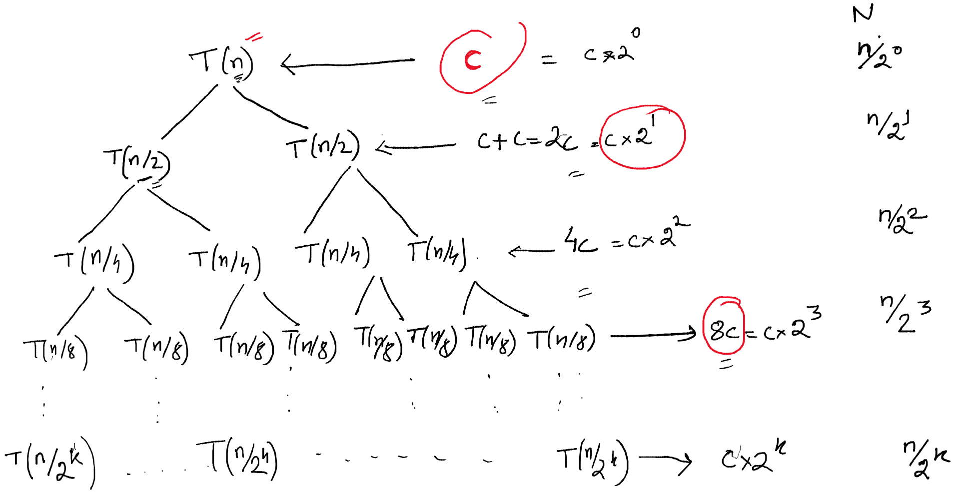
Recursion tree method

$$T(n) = \begin{cases} 2T(n/2) + c, & n > 1 \\ c, & n = 1 \end{cases}$$



Recursion stop $\rightarrow \frac{n}{2^k} = 1$

$\Rightarrow n = 2^k$

$\Rightarrow \log_2 n = \log_2(2^k)$

$\Rightarrow \log_2 n = k \underline{\log_2 2}$

$\Rightarrow k = \log_2 n$

GP sum $= \frac{a(r^n - 1)}{r-1}$

Total cost

$c + 2c + 4c + 8c + \dots\dots$

$+ 2^k \times c$

$= c2^0 + c2^1 + c2^2 + c2^3 + \dots$

$c2^k$

$= c(2^0 + 2^1 + \dots + 2^k)$

$= c \times \frac{1 \times (2^{k+1} - 1)}{2-1}$

$= c \times 2^k \cdot 2 - c$

$= c \cdot 2^k - c$

$\simeq 2^k$

$= 2^{\log_2 n}$

$= n^{\log_2 2}$

$= n$

$\Rightarrow TC = O(n)$.

$$T(n) = \begin{cases} 2T(n/2) + n, & n > 1 \\ 1, & n = 1 \end{cases}$$

$T(n) \longrightarrow n \longrightarrow$ 0th step

$T(n/2) \quad T(n/2) \longrightarrow n/2 + n/2 = n \longrightarrow$ 1st step

$T(n/4) \quad T(n/4) \quad T(n/4) \quad T(n/4) \longrightarrow n/4 + n/4 + n/4 + n/4 = n$ — 2nd

$T(n/8) \quad T(n/8) \quad T(n/8) \quad T(n/8) \quad T(n/8) \quad T(n/8) \quad T(n/8) \quad T(n/8) = 8 \times n/8 = n$ — 3rd

$\vdots$

$T(n/2^k) \quad T(n/2^k)$ $\qquad n \longrightarrow$ k^{th} step

n
$n/2^1$
$n/2^2$
$n/2^3$
$n/2^k$

Recursion stop, $n/2^k = 1$

$\Rightarrow n = 2^k$

$\Rightarrow \log_2 n = \log_2 (2^k)$

$\Rightarrow \log_2 n = k \times \log_2 2$

Total cost $= n \times k = n \times \log_2 n$

$T.C = O(n \log_2 n)$