

2. Outline the time analysis of the following recursive programs using recursion tree method for

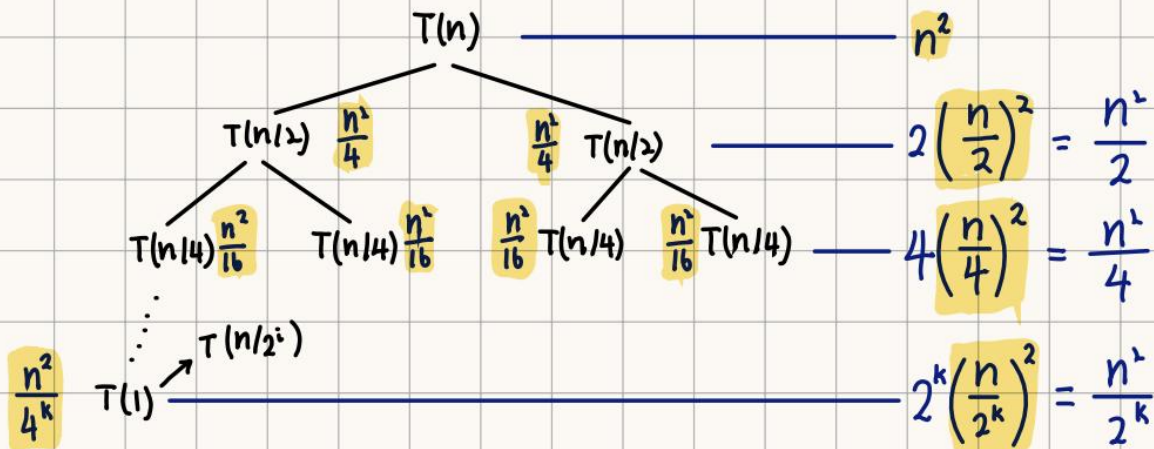
i) $T(n) = 2T(n/2) + n^2$; where $n > 1$

ii) $T(n) = T(n/3) + T(2n/3) + n$; where $n > 1$

■ Work Done

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

Work Done



$\frac{n}{2^k} = 1 \rightarrow n = 2^k \rightarrow k = \log_2 n$

$k \rightarrow$ level of tree

Total work = $n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \dots + \frac{n^2}{2^k}$
 $= n^2 \left(\frac{1}{2^0} + \frac{1}{2^1} + \frac{1}{2^2} + \dots + \frac{1}{2^k} \right)$ } Inf-nite GP.

$a = n^2$

$S = a \left(\frac{1-r^{k+1}}{1-r} \right)$

$r = \frac{1}{2}$

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

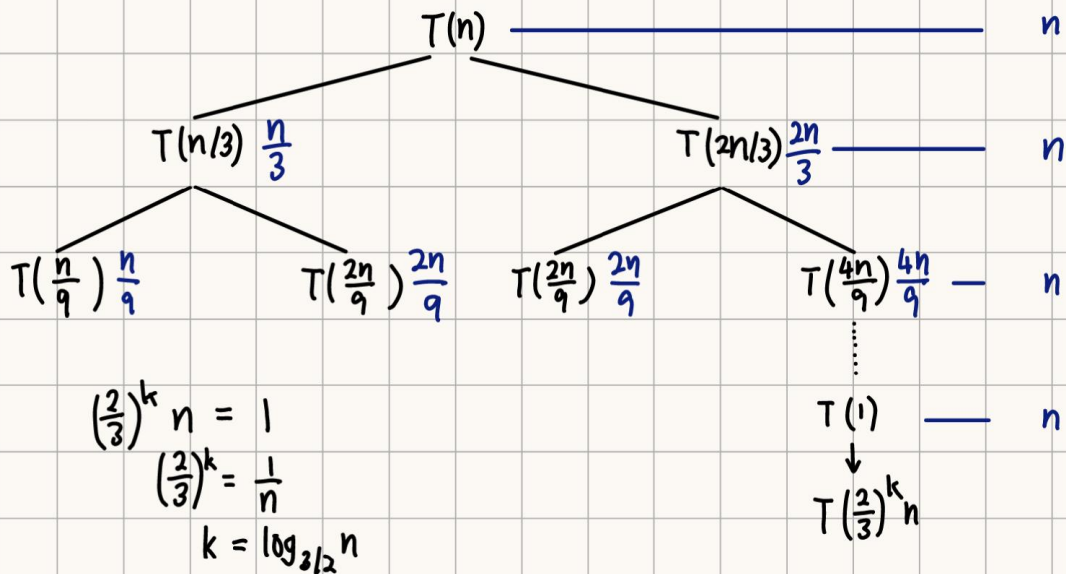
$k = \log_2 n + 1$

$= n^2 \left(2 \left(1 - \frac{1}{2^{n+1}} \right) \right)$

\therefore Running time complexity = $O(n^2)$

$$(ii) T(n) = \begin{cases} T(n/3) + T(2n/3) + n & ; n > 1 \\ 1 & ; n = 1 \end{cases}$$

Work Done



$$\begin{aligned} \text{Total Work} &= n \times k \\ &= n \times \log_{3/2} n \end{aligned}$$

$$\therefore \text{Running time complexity} = O(n \log(n))$$