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

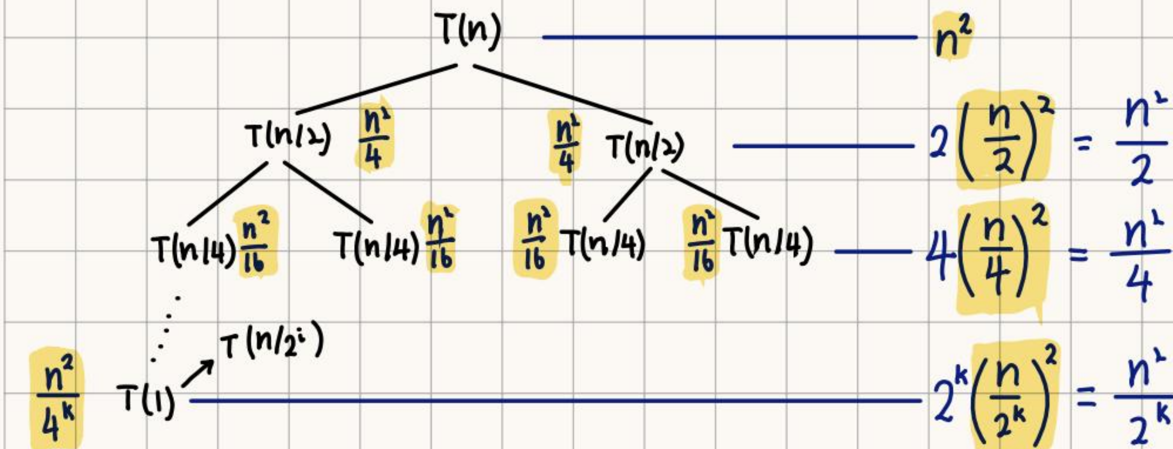
   i) $T(n) = 2T(n/2) + n^2$; where n>1

   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



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

k → level of tree

$$\text{Total work} = n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \dots + \frac{n^2}{2^k}$$

$$= n^2\left(\frac{1}{2^0} + \frac{1}{2^1} + \frac{1}{2^2} + \dots \frac{1}{n}\right) \Big\}\ \text{Infinite GP.}$$

$a = n^2$ $\qquad S = a\left(\frac{1-r^k}{1-r}\right)$

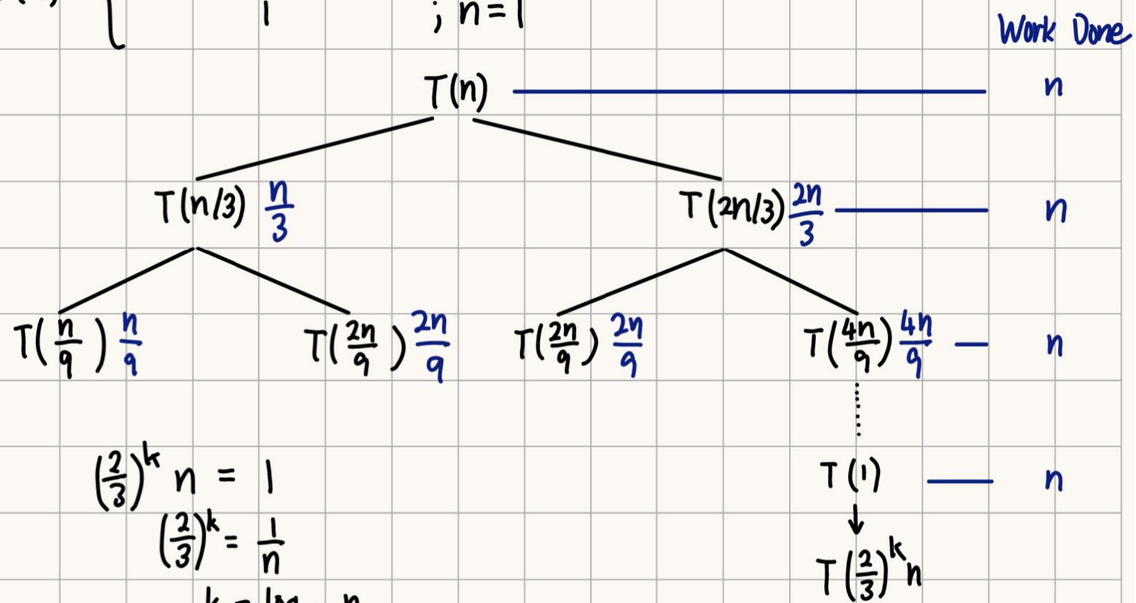
$r = \frac{1}{2}$ $\qquad = n^2\left(\frac{1-1/2^{\log_2 n+1}}{1-1/2}\right)$

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

∴ Running time complexity = $O(n^2)$

(ii)

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



$$\left(\frac{2}{3}\right)^k n = 1$$

$$\left(\frac{2}{3}\right)^k = \frac{1}{n}$$

$$k = \log_{3/2} n$$

Total Work $= n \times k$

$= n \times \log_{3/2} n$

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