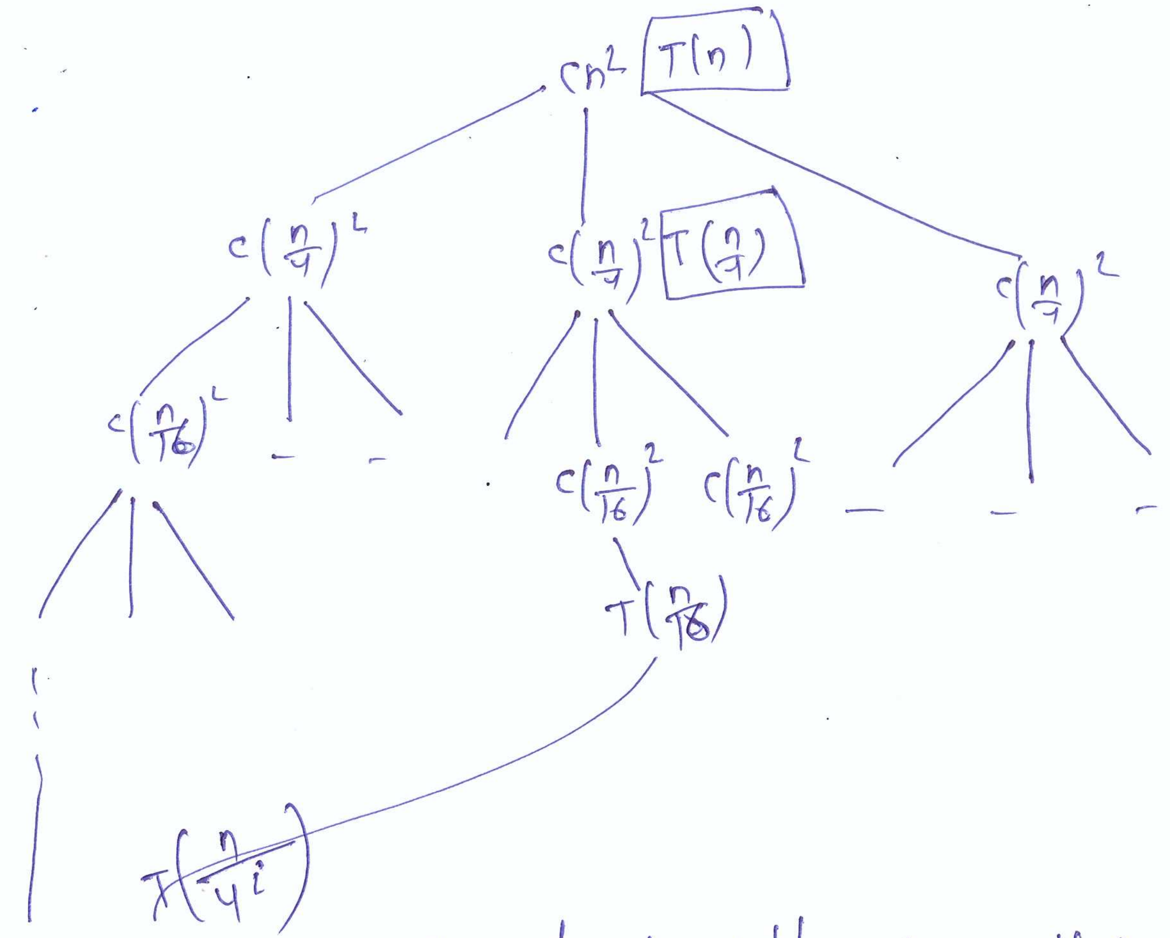
$T(n) = 3T(n/4) + cn^2$



At the end the problem size will be 1

No of leaf node = ?

$\frac{n}{4^i} = 1 \Rightarrow 4^i = n$

$\log_4 4^i = \log_4 n$

$\Rightarrow \boxed{i = \log_4 n}$

thus no of leaf node $= 3^i = 3^{\log_4 n}$

$\boxed{= n^{\log_4 3}}$

$T(n) = cn^2 + 3c\left(\frac{n}{4}\right)^2 + 9c\left(\frac{n}{16}\right)^2 + \text{———} + n^{\log_{64}^{9}} c\left(\frac{n}{4^{n\log_4^3}}\right)^2$

$= cn^2 + \frac{3}{16}cn^2 + \frac{9}{256}cn^2 + \text{———}$

$= cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \text{———}$

no of subproblem

(2b) $T(n) = cn^2 + 3c\left(\frac{n}{4}\right)^2 + 3^2 c\left(\frac{n}{16}\right)^2 + 3^3 c\left(\frac{n}{64}\right)^2$

$+ \text{———} + n^{\log_4^3} c\left(\frac{n}{4^{\log_4^n}}\right)^2$

$= \text{———} + n^{\log_4^3} c\left(\frac{n}{4^{\log_4^n}}\right)$

$= \text{———} + n^{\log_4^3} c(1)$

$= \sum_{i=0}^{\log_4^n - 1} \left(\frac{3}{16}\right)^i cn^2 + n^{\log_4^3}$

$=$ ~~Solve this~~

$T(n) = aT(n/b) + f(n)$

$a \geq 1 \quad b > 1$ Reason ?

$f(n)$ is asymptotically positive function

(i) $f(n) = O(n^{\log_b a - \varepsilon})$ for some constant $\varepsilon > 0$ then $T(n) = \Theta(n^{\log_b a})$

Example

$T(n) = 9T(n/3) + n$

$a = 9, b = 3, f(n) = n, n^{\log_b a} = n^{\log_3 9} = n^2$

$f(n) = O(n^{\log_3 9 - \varepsilon})$ when $\varepsilon = 1$

So $\boxed{T(n) = \Theta(n^2)}$

$T(n) = 8T(n/2) + \Theta(n^2)$

$a = 8, b = 2, f(n) = \Theta(n^2) \quad n^{\log_b a} = n^{\log_2 8} = n^3$

$f(n) = O(n^{\log_2 8 - \varepsilon})$ for $\varepsilon = 1$

$\boxed{T(n) = \Theta(n^3)}$

$T(n) = 7T(n/2) + \Theta(n^2)$

$a = 7 \quad b = 2 \qquad f(n) = \Theta(n^2) \qquad n^{\log_b^a} = n^{\log_2^7}$

$= n^{2.80\ldots}$

$f(n) = O(n^{\log_2^7 - \varepsilon})$ for $\varepsilon = .80\ldots$

so $\boxed{T(n) = \Theta(n^{\log_2^7}) = \Theta(n^{2.80\ldots})}$

(ii) $f(n) = \Theta(n^{\log_b^a})$ then $T(n) = \Theta(n^{\log_b^a} \log n)$

$T(n) = 2T(n/2) + \Theta(n) \longrightarrow$ Merge Sort

$a = b = 2 \quad f(n) = \Theta(n) \qquad n^{\log_b^a} = n^{\log_2^2} = n$

$f(n) = \Theta(n^{\log_2^2}) = \Theta(n)$

so

$T(n) = \Theta(n^{\log_2^2} \log n)$

$\boxed{= \Theta(n \log n)}$

Same as Tree Method

(iii) $f(n) = \Omega(n^{\log_b^a + \epsilon})$ for some constant $\epsilon > 0$ and if

$a f(n/b) \leq c f(n)$ for $c < 1$ and sufficiently large n

$\xrightarrow{}$ Constant

$$\boxed{T(n) = \Theta(f(n))}$$

$T(n) = 3T(n/4) + n \log n$

$a = 3 \quad b = 4 \quad f(n) = n \log n \quad n^{\log_b^a} = n^{\log_4^3} = n^{.793}$

$f(n) = \Omega(n^{.793 + \epsilon})$ when $\epsilon \simeq 0.2$

So (iii) case ~~can be~~ applied but check

$a f(n/b) \leq c f(n)$

$3 f(n/4) \leq c f(n)$

$3 \left(\frac{n}{4}\right) \log \left(\frac{n}{4}\right) \leq c n \log n$

$\frac{3}{4} n \log n - \frac{3}{4} n \log 4 \leq c n \log n$

for $c = 3/4$ this will hold so

$$\boxed{T(n) = \Theta(n \log n)}$$

(i) $f(n) = O\left(n^{\log_b^a - \epsilon}\right)$

↳ $f(n)$ be POLYNOMIALLY SMALLER than $n^{\log_b^a}$ by a factor of n^{ϵ}

(ii) $f(n) = \Theta\left(n^{\log_b^a}\right)$

(iii) $f(n) = \Omega\left(n^{\log_b^a + \epsilon}\right)$

↳ $f(n)$ be POLYNOMIALLY LARGER than $n^{\log_b^a}$ by a factor of n^{ϵ}

(i)

(ii) → Gap $f(n)$ is smaller but not polynomially

(ii) → Gap $f(n)$ is larger but not polynomiall

(iii)

$T(n) = 2T(n/2) + n\log n$

$a = b = 2$ $\quad f(n) = n\log n$ $\quad n^{\log_b^a} = n^{\log_2^2} = n$

Now f(n) is larger than $n^{\log_b^a}$

But it is not POLYNOMIALLY larger

i.e. $\dfrac{f(n)}{n^{\log_b^a}} = \dfrac{n\log n}{n} = \log n$

not like n^{ε}

So its in Gap b/w 2 and 3