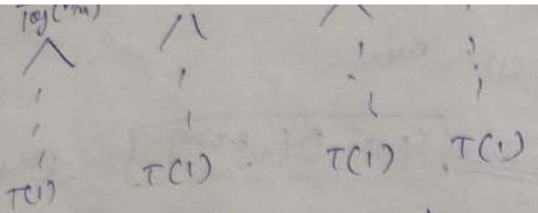


No. of leaf nodes $= 2^i$

we get leaf node when, $\frac{n}{2^i} = 1 \Rightarrow n = 2^i \Rightarrow \log_2 n = i$

$$T(n) = \frac{n}{\log n} + \frac{n}{\log(n/2)} + \frac{n}{\log(n/4)} + \dots + \frac{n}{\log\frac{n}{2^i}} + 2^i \cdot T(1)$$

$$= n\left(\frac{1}{\log 2^i} + \frac{1}{\log\frac{2^i}{2}} + \frac{1}{\log\frac{2^i}{2^2}} + \dots + \frac{1}{\log\frac{2^i}{2^{i-1}}}\right] + n\cdot\theta(1)$$

$$= n\left[\frac{1}{i} + \frac{1}{i-1} + \dots + \frac{1}{2} + 1\right] + n\cdot\theta(1)]$$

$$= n.\log i + n\theta(1).$$

$$= n.\log\log n + n\cdot\theta(1)$$

$$T(n) = \theta(n\log\log n)$$

Q. Solve the recurrence

$T(n) = T$ ...

when $T(0) = 0$

04/08/16

. Substitution method

It consists of 2 main steps

(i) Guess the solution

(ii) use mathematical induction to ... guess is correct.

How to guess?-

↳ Draw a recursion tree to gen...

↳ Apply master method.

Solve the recurrence:

$T(n) = 2T(\frac{n}{2}) + n$ ...

$= cn + cn \ldots$ (times)

$= cn + \log_{3/2} n \; cn$

$$\boxed{T(n) = \Theta(n \log_{3/2} n)}$$

Q.

$T(n) = T(n/2) + T(n/4) + T(n/8) + n$

Sol^n: $T(n) = T(n/2) + T(n/4) + T(n/8) + cn$

$T(n)$:

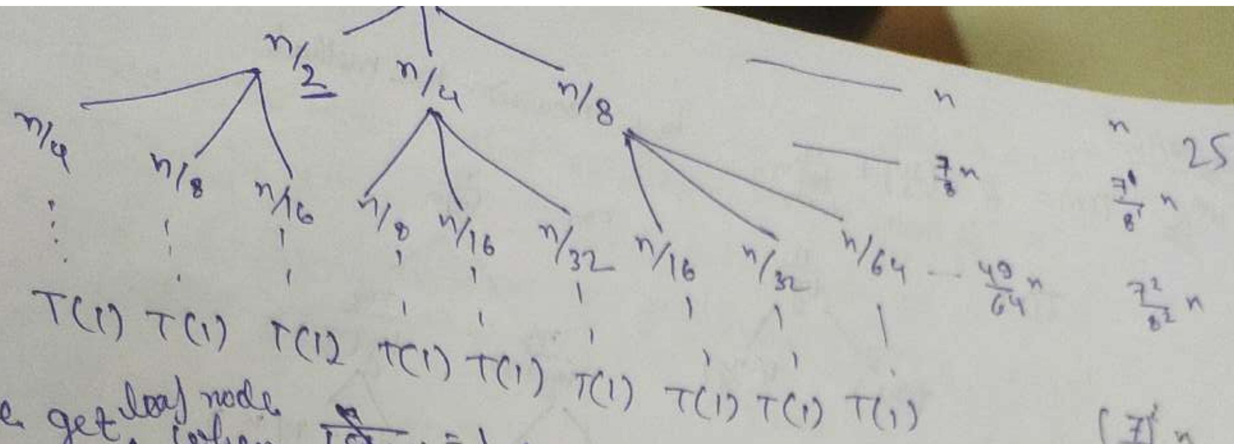
cn

$T(n/2) \quad T(n/4) \quad T(n/8)$

n

$n/2 \quad n/4 \quad n/8$

$T\left(\frac{n}{4}\right) \; T\left(\frac{n}{8}\right) \; T\left(\frac{n}{16}\right) \; T\left(\frac{n}{8}\right) \; T\left(\frac{n}{16}\right) \; T\left(\frac{n}{32}\right) \; T\left(\frac{n}{16}\right) \; T\left(\frac{n}{32}\right) \; T\left(\frac{n}{6\ldots}\right)$



we get leaf node when ~~$\left(\frac{8}{7}\right)^i = 1$~~

$\frac{n}{2^i} = 1$, $n = 2^i$

$i = \log_2 n$

~~$\Rightarrow n = \left(\frac{8}{7}\right)^i$~~

~~$\Rightarrow \log_{8/7} n = i$~~

total no. of leaf nodes $= 3^i$

total cost $T(n) = n + \frac{7}{8}n + \frac{7^2}{8^2}n + \cdots + \left(\frac{7}{8}\right)^{i-1} n + T(1)\cdot 3^i$

$$T(n) = n\sum_{i=0}^{i-1}\left(\frac{7}{8}\right)^i + T(1)\cdot 3^{\log_2 n}$$

$$T(n) \leq n\sum_{i=0}^{\infty}\left(\frac{7}{8}\right)^i + T(1)\cdot n^{\log_2 3}$$

~~$T(n) \leq n\cdot\left(\frac{1}{1-\frac{7}{8}}\right) + T(1)\cdot n^{\log_{8/7} 3}$~~

~~$T(n) \leq n\cdot 8 + T(1)\cdot n^{\log_{8/7} 3}$~~

~~$\boxed{T(n) = O(n^{\log_{8/7} 3})}$~~ ~~Answer~~ ~~$\boxed{T(n) = O(n^{8.227})}$~~ $O(n)$

$$T(n) \leq n\cdot\left(\frac{1}{1-7/8}\right) + T(1)\cdot n^{\log_2 3}$$

$$T(n) \leq 8n + \Theta(1)\cdot n^{1.585}$$

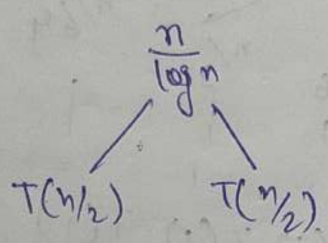
$\therefore$ $$\boxed{T(n) = O(n^{1.585})}$$

Answer

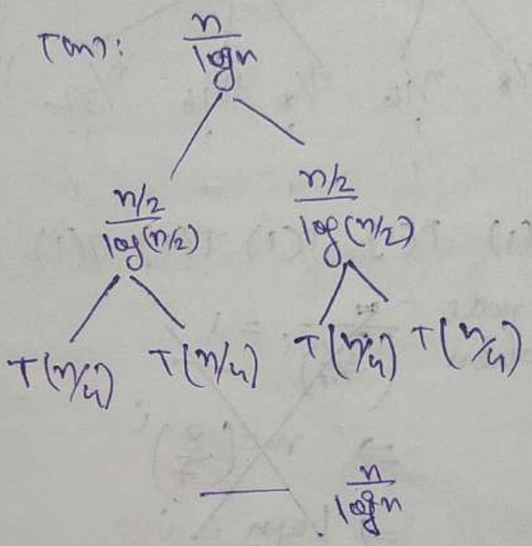
3/08/16

10m Q. $T(n) = 2T\left(\frac{n}{2}\right) + \frac{n}{\log n}$ by recursion-tree method.

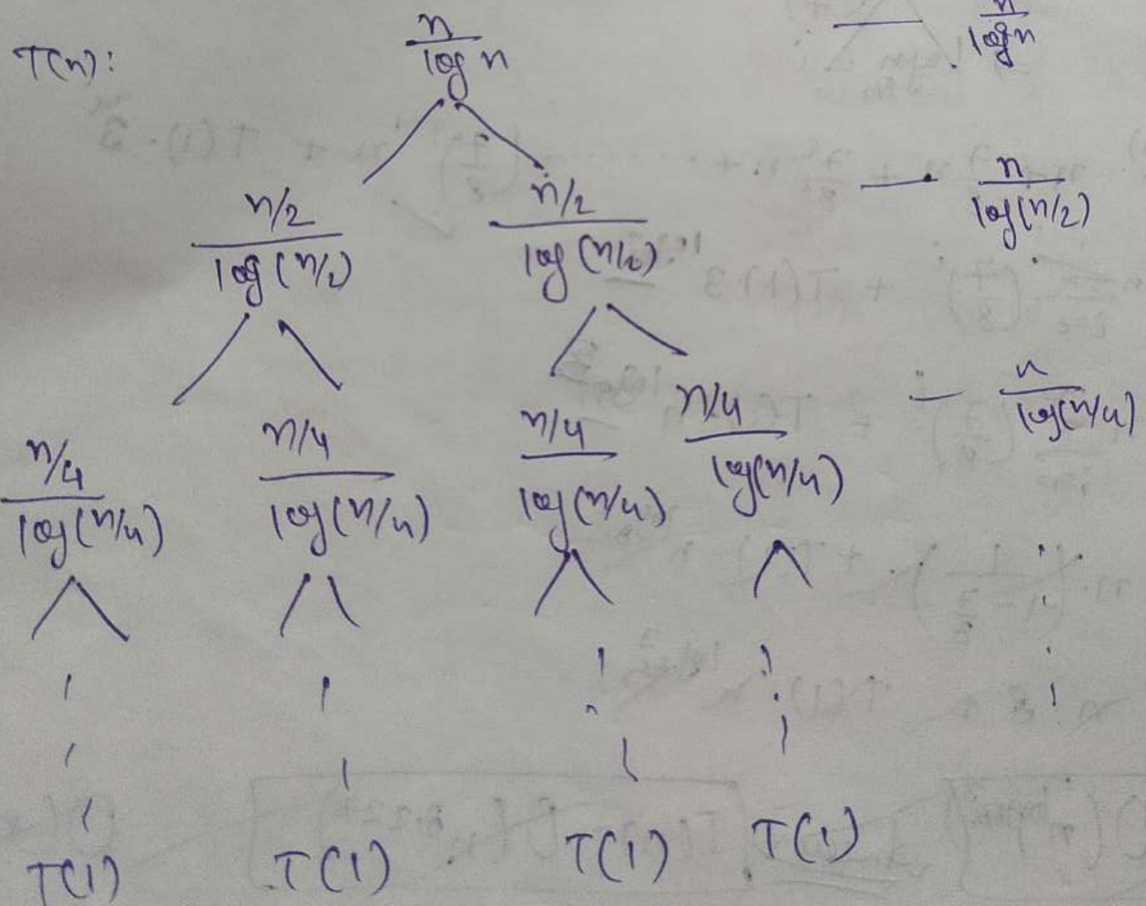
T(n):



T(n):



T(n):



No. of leaf nodes $= 2^i$

We get leaf node when, $\frac{n}{2^i} = 1 \Rightarrow n = 2^i \Rightarrow \log_2 n$

$+ \frac{n}{n} + 2^i$