Sub :

Day

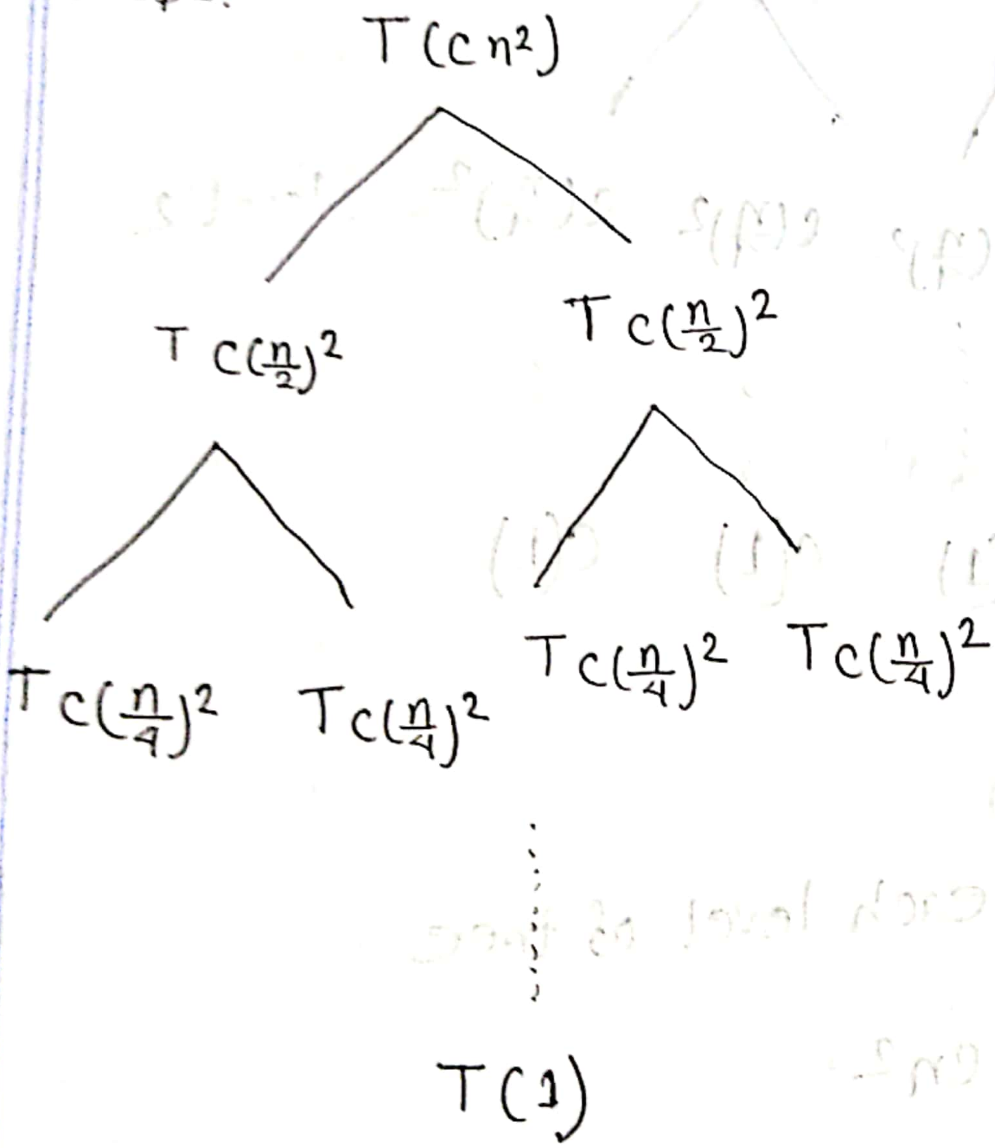
Time : Date : / /

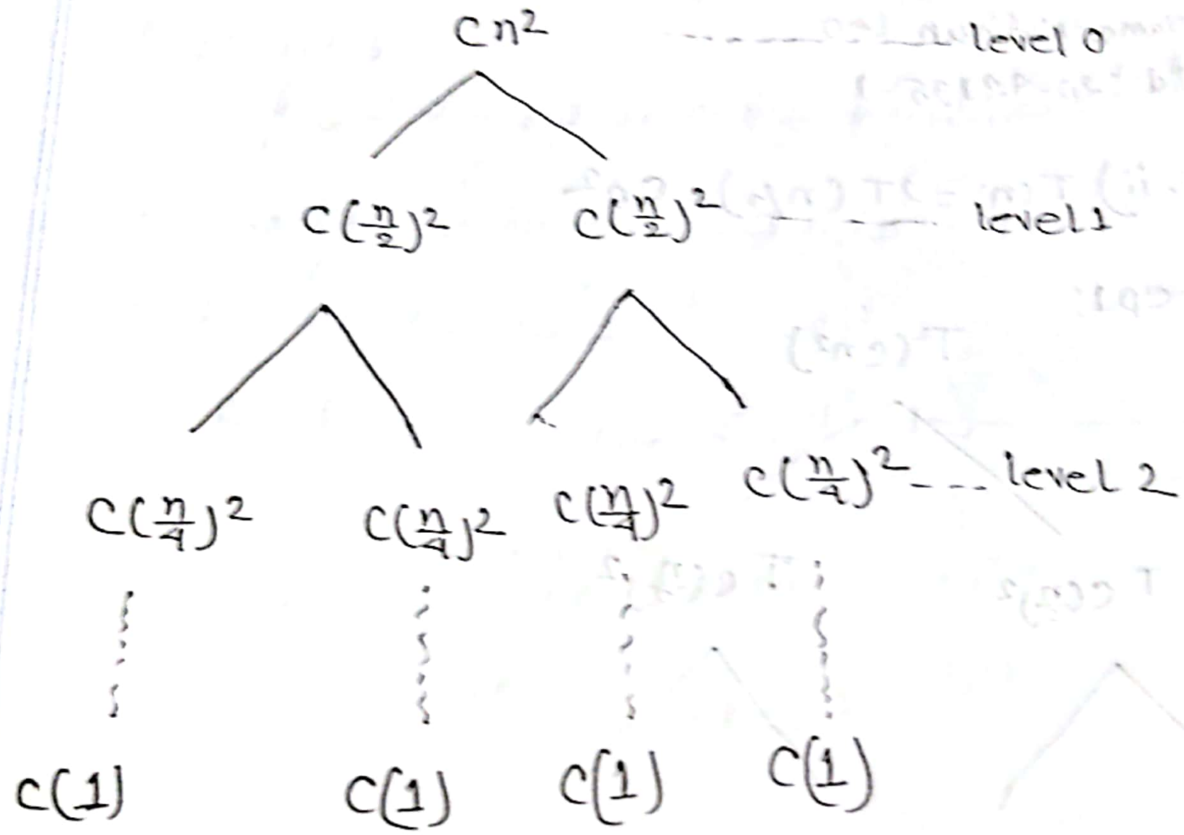
Name: Nafinun Leo

Id : 20-42195-1

1. ii) $T(n) = 2T(n/2) + cn^2$

Step 1:





Step 2, Part 1:

Cost of each level of tree

$$\text{level } 0 = cn^2$$

$$\text{level } 1 = c\left(\frac{n}{2}\right)^2 \times 2 = \frac{1}{2}cn^2$$

$$\text{Level } 2 = c\left(\frac{n}{4}\right)^2 \times 4 = \frac{1}{4}cn^2$$

Sub :

Step2; Part 2:

Determine total number of levels:

size of sub-problems

at level-0 $=\left(\frac{n}{2^0}\right)^2$

at level 1 $=\left(\frac{n}{2^1}\right)^2$

at level 2 $=\left(\frac{n}{2^2}\right)^2$

Continuing in similar manner

at level i $=\left(\frac{n}{2^i}\right)^2$

Suppose last level is x

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

$$\Rightarrow n = 2^x$$

$$\Rightarrow x = \log_2 n$$

So, the total number of levels in tree $\log_2 n + 1$

Step 2; Part 3:

Determine number of nodes in last level

in level 0, we have $2^0 = 1$ node

in level 1, we have $2^1 = 2$ nodes

in level 2, we have $2^2 = 4$ nodes

Continuing in similar manner we have

in level $\log_2 n$, we have $= 2^{\log_2 n}$

$= n^{\log_2 2}$

$= n$

Step 2; Part 4:

Determing the total cost of last level:

$n \times T(1)$

$= n \times 1$

$= n$

$= O(n)$

Step 3:

Now, we add cost of all levels:

$$T(n) = \{cn^2 + \tfrac{1}{2}cn^2 + \tfrac{1}{4}cn^2 + \cdots\} + O(n)$$

For $\log_2 n$ levels

$$T(n) = \sum_{i=0}^{\log_2 n - 1} \left(\tfrac{1}{2}\right)^i cn^2 + O(n)$$ [$1 + \frac{1}{2} + \left(\frac{1}{2}\right)^2 + \cdots$ forms an infinite geometric progression]

$$= \frac{1}{1-\frac{1}{2}} cn^2 + O(n)$$

$$= 2cn^2 + O(n)$$
$$= O(n^2)$$

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

(Ans:)

2. i) $T(n) = 4T(n/2) + n$

Here,
$a = 4$

$b = 2$

$f(n) = n$

Now, $n^{\log_b n} = n^{\log_b a}$

$$\Rightarrow n^{\log_b a} = a^{\log_b n}$$

$$= n^{\log_b a}$$

$$= n^{\log_2 4}$$

$$= n^2$$

Here, $f(n) < n^{\log_2 4}$

So, it matches with cas 1.

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

(Ans:)