```
int factorial(int n)
{
    if (n <= 1) {
        return 1;
    }
    return n * factorial(n-1);
}
```

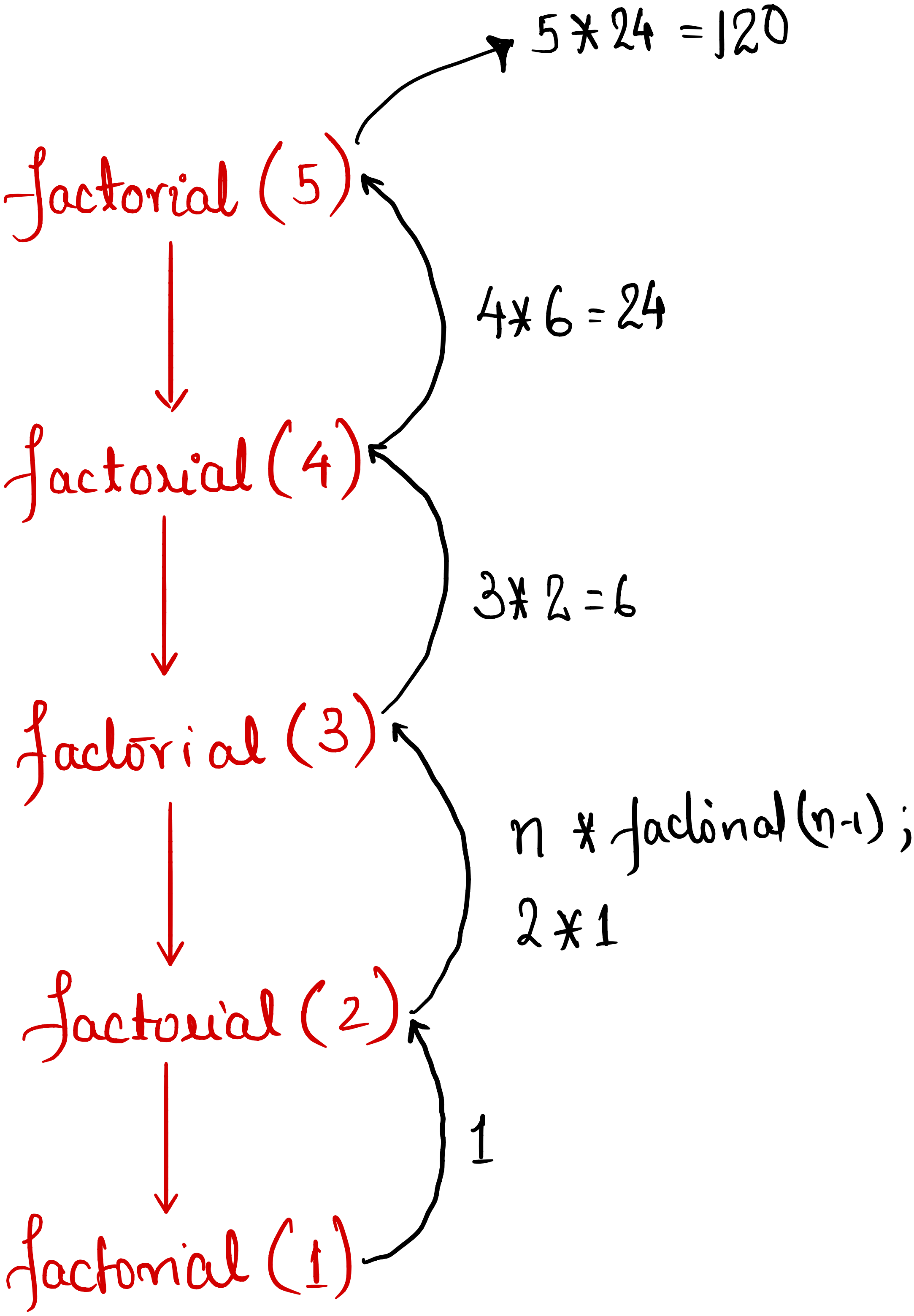
(Termination Condition)

Dividing

Composing

Recursion :-

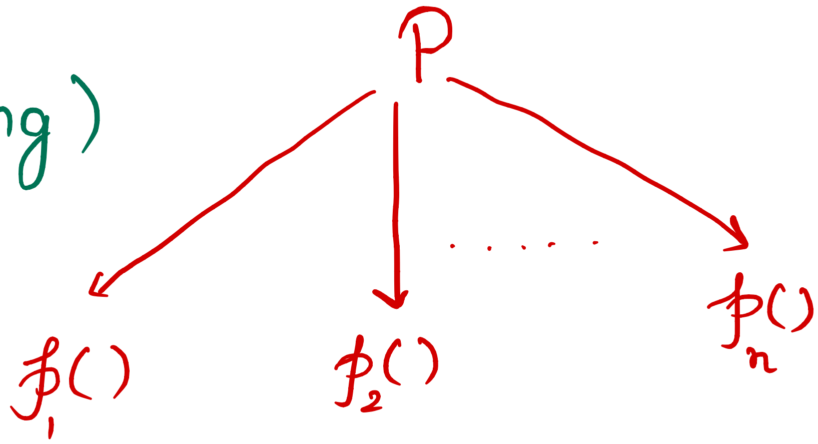
A process can be described in terms of processes of relatively smaller size.



5 * 24 = 120
factorial (5)
4 * 6 = 24
factorial (4)
3 * 2 = 6
factorial (3)
n * factorial (n-1) ;
2 * 1
factorial (2)
1
factorial (1)

Recursion can be used as a programming paradigms.

(Dividing)



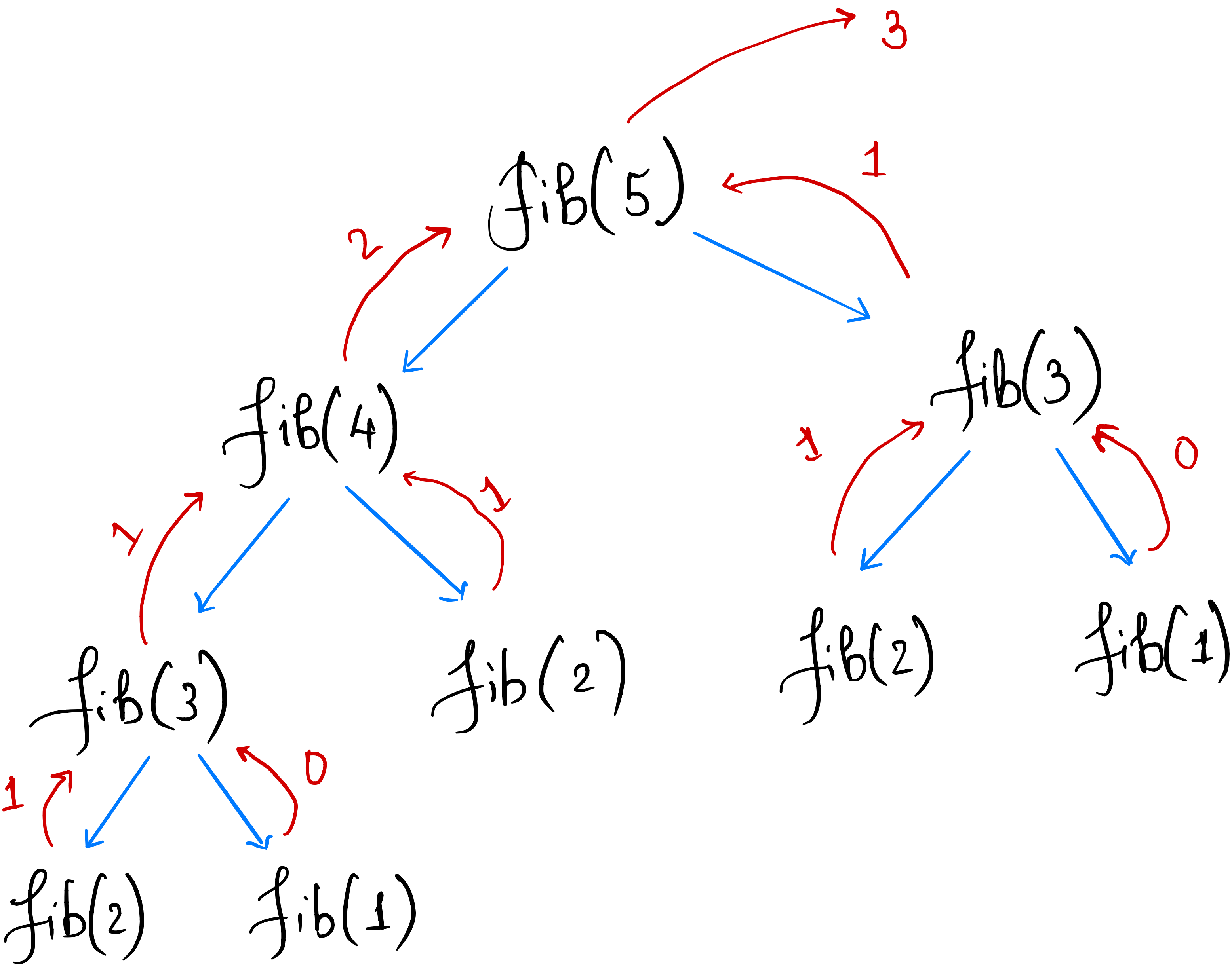
$p_1(), p_2() \dots p_n()$ are having same structure as that of P.

(Composing) How to compose the results from smaller subproblems to solve the ultimate or bigger problem (P)

(Termination) (Base) During the process of dividing the problems at some place we will realize that subproblem is small enough to solve and since we already have the solution for that we stop.

Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, . . .



Recurrence Relation.

A recurrence relation is an equation or an inequality that describes a function in terms of its value on smaller inputs.

Example :-

Let P be a problem to be solved.

A recursive algorithm might divide P into smaller subproblems of unequal lengths

For example :-

n size problem P

- $\frac{n}{4}$ sized-Problem P_1 of same structure as that of P
- $\frac{n}{2}$ sized-Problem P_2 of same structure as that of P
- another $\frac{n}{4}$ sized problem has same structure

Each recursive call is atributed by dividing the problem into subproblems and also combining the results of these subproblems.
Let the time complexity of this division and combination is $\Theta(f(n))$.

Then,

$$T(n) = T\left(\frac{n}{2}\right) + 2T\left(\frac{n}{4}\right) + \Theta(f(n))$$

Our Interest In this topic is,

Solving Recurrence Relations.

Substitution Method. (Not Included, Revise).

Iteration Method.

1. Recursion Tree Method
2. Master Theorem

Iteration Method

Find the time complexity of algorithm described by the following recurrence relation.

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n-1) + n & \text{if } n > 1 \end{cases}$$

$$T(n) = T(n-1) + n$$

$$= \big(T(n-2) + (n-1)\big) + n$$

$$= T(n-3) + (n-2) + (n-1) + n$$

$$= T(n-4) + (n-3) + (n-2) + n$$

$$\vdots$$

$$= T(1) + 2 + 3 + 4 + \cdots + (n-3) + (n-2) + n$$

$$= \frac{n(n+1)}{2}$$

$$T(n) = O(n^2) \qquad (T(n) \text{ is } \Theta(n^2))$$

Example

$$T(n) = \begin{cases} 2T(n-1) - 1, & \text{if } n > 0 \\ 1 & \text{otherwise} \end{cases}$$

$$T(n) = 2T(n-1) - 1$$

$$= 2\left(2T(n-2) - 1\right) - 1$$

$$= 2^2 T(n-2) - 2 - 1$$

$$= 2^2\left(2T(n-3) - 1\right) - 2 - 1$$

$$= 2^3 T(n-3) - 2^2 - 2 - 2^0$$

$$\vdots$$

$$= 2^{n-1}T(1) - 2^{n-2} - 2^{n-3} - 2^{n-4} \dots - 2^1 - 2^0$$

(Recall :- Sum of first n terms of a GP $a, ar, ar^2, \dots, ar^{n-1}$ is $\dfrac{a(r^n - 1)}{(r-1)}$ $r \geq 1$)

$$T(n) = 2^{n-1} - \left(2^0 + 2^1 + 2^2 + \cdots + 2^{n-2}\right)$$

$$T(n) = 2^{n-1} - \frac{1 \cdot (2^{n-1} - 1)}{(2-1)}$$

$$T(n) = \cancel{2^{n-1}} - \cancel{2^{n-1}} + 1$$

$$T(n) = 1$$

$$T(n) = \Theta(1) \left(\text{or } O(1)\right)$$

Example:-

$$T(n) = \begin{cases} 3T(n-1) & \text{, if } n>0 \\ 1 & \text{, otherwise} \end{cases}$$

$$\begin{aligned} T(n) &= 3T(n-1) \\ &= 3(3T(n-2)) \\ &= 3^2(T(n-2)) \\ &\vdots \\ &= 3^n(T(n-n)) \\ &= 3^n T(0) \\ &= 3^n \end{aligned}$$

$T(n) = 3^n$

$\leq c3^n, c>1$

$T(n) = 3^n$

$\geq c\,3^n, 0<c<1$

$$T(n) = \Theta(3^n)$$

Example :-

$$T(n) = \begin{cases} 3T(n/4) + n & n > 1 \\ 1 & n = 1 \end{cases}$$

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

$$= 3\left(3T\left(\frac{n}{4^2}\right) + \frac{n}{4}\right) + n$$

$$= 3^2 T\left(\frac{n}{4^2}\right) + \frac{3}{4}n + n$$

$$= 3^2\left(3T\left(\frac{n}{4^3}\right) + \frac{n}{4^2}\right) + \frac{3}{4}n + n$$

$$= 3^3 T\left(\frac{n}{4^3}\right) + \frac{3^2}{4^2}n + \frac{3}{4}n + n$$

We have to continue doing this kind of substitution till size of problem $\left(\frac{n}{4^3}\right)$ in above equation become 1.

Let At k^{th} substitution

$$\frac{n}{4^k} = 1$$

i.e., $n = 4^k$

$$\log_4 n = \log_4 4^k$$

$$\log_4 n = k$$

$$\boxed{k = \log_4 n}$$

$$T(n) = 3^{\log_4 n}\, T\left(\frac{\cancel{n}}{\cancel{n}}1\right) + \left(\frac{3}{4}\right)^{\log_4 n - 1} \cdot n +$$

$$\left(\frac{3}{4}\right)^{\log_4 n - 2} \cdot n + \dots + \left(\frac{3}{4}\right)^2 n + \left(\frac{3}{4}\right) n + n$$

$$T(n) = 3^{\log_4 n} + n\left(1 + \left(\frac{3}{4}\right) + \left(\frac{3}{4}\right)^2 + \dots\dots + \right.$$

$$\left. \left(\frac{3}{4}\right)^{\log_4 n - 2} + \left(\frac{3}{4}\right)^{\log_4 n - 1}\right)$$

$$T(n) = 3^{\log_4 n} + n\left(\frac{1-(3/4)^{\log_4 n}}{(1-3/4)}\right)$$

(Sum of n terms of a G.P : $a, ar, ar^2, ar^3, \dots, ar^{n-1} = a\left(\frac{1-r^n}{1-r}\right)$ where $|r|<1$)

$$T(n) = 3^{\log_4 n} + 4n\left(1 - \frac{3^{\log_4 n}}{n}\right)$$

$$= 3^{\log_4 n} + 4n - 4.3^{\log_4 n}$$

$$= 4n - 3.3^{\log_4 n}$$

$$= 4n - 3n^{\log_4 3} \qquad (a^{\log_x b} = b^{\log_x a})$$

$$T(n) = 4n - 3n^{0.79} \qquad (\log_4 3 \approx 0.79)$$

$$\leq 4n \qquad (\forall n \geq n_0)$$

$T(n)$ is $O(n)$.

||ly $T(n) \geq 4n - 3n$
$= n \qquad (\forall n \geq n_0)$

$T(n)$ is $\Omega(n)$

$\Rightarrow T(n)$ is $\Theta(n)$

Example: (factorial)

$$T(n) = \begin{cases} T(n-1) + 1 & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

$$\begin{aligned} T(n) &= T(n-1) + 1 \\ &= T(n-2) + 2 \\ &= T(n-3) + 3 \\ &\vdots \\ &= T(1) + (n-1) \\ &= n \end{aligned}$$

$$T(n) = n$$

$$T(n) = \Theta(n)$$

Recursion-Tree Method

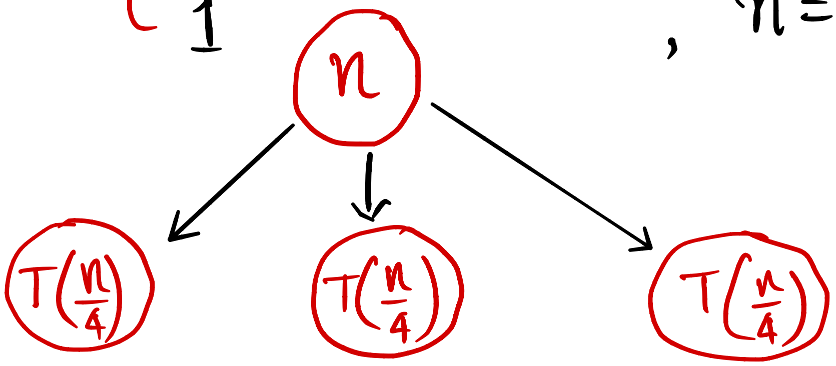
In recursion tree, each node represents the cost of single sub-problem somewhere in the set of recursive function invocations. (CLRS)

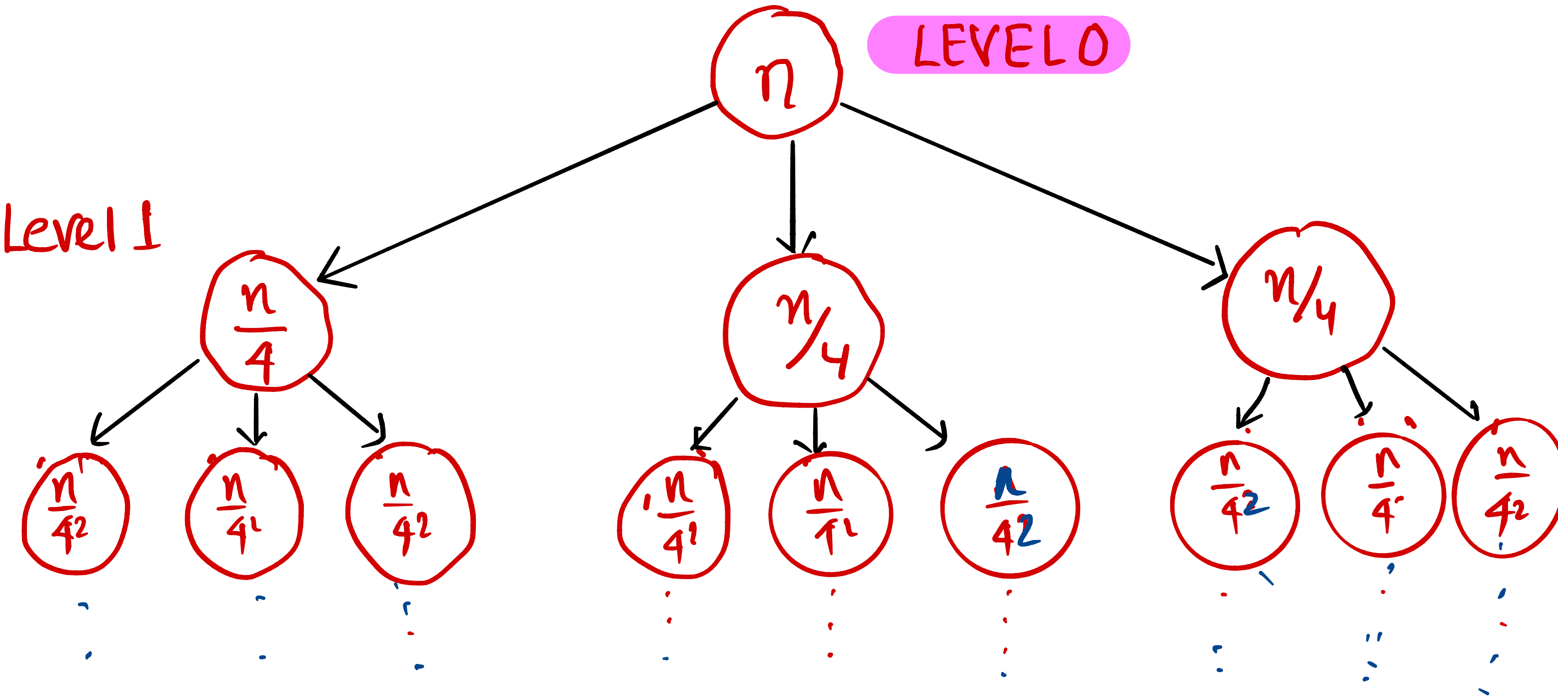
Inorder to find total Cost.

(1) Sum the costs at each level of tree. to find the per-level costs.

(2) Sum the per-level costs to determine all levels cost.

$$T(n) = \begin{cases} 3T\left(\frac{n}{4}\right) + n & , \ n > 1 \\ 1 & , \ n = 1 \end{cases}$$





$$n = 4^k$$

$$k = \log_4 n$$

Problem Size

Levels	No. of Nodes	Cost/Node	Total Cost
0	1	n	n
1	3	$\frac{n}{4}$	$\frac{3}{4}n$
2	3^2	$\frac{n}{4^2}$	$\frac{3^2}{4^2}n$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\log_4 n - 1$	$3^{\log_4 n - 1}$	$\frac{n}{4^{\log_4 n - 1}}$	$\left(\frac{3}{4}\right)^{\log_4 n - 1} n$
$\log_4 n$	$3^{\log_4 n}$	1	$3^{\log_4 n}$

$$T(n) = n + \frac{3}{4}n + \frac{3^2}{4^2}n + \cdots + \left(\frac{3}{4}\right)^{\log_4 n - 1} n + 3^{\log_4 n}$$

$$= n\left(1 + \left(\frac{3}{4}\right) + \left(\frac{3}{4}\right)^2 + \left(\frac{3}{4}\right)^3 + \cdots + \left(\frac{3}{4}\right)^{\log_4 n - 1}\right) + 3^{\log_4 n}$$

$$= n\left(\frac{1 - 3^{\log_4 n}/n}{1 - 3/4}\right) + 3^{\log_4 n}$$

$$= 4n\left(1 - \frac{3^{\log_4 n}}{n}\right) + 3^{\log_4 n}$$

$$= 4n - \frac{4n\,3^{\log_4 n}}{n} + 3^{\log_4 n}$$

$$= 4n - 4.3^{\log_4 n} + 3^{\log_4 n}$$

$$= 4n - 3\,3^{\log_4 n}$$

$$\left(3^{\log_4 n} = n^{\log_4 3}\right)$$

$$T(n) = 4n - 3n^{\log_4 3}$$

$$= 4n - 3n^{0.79}$$

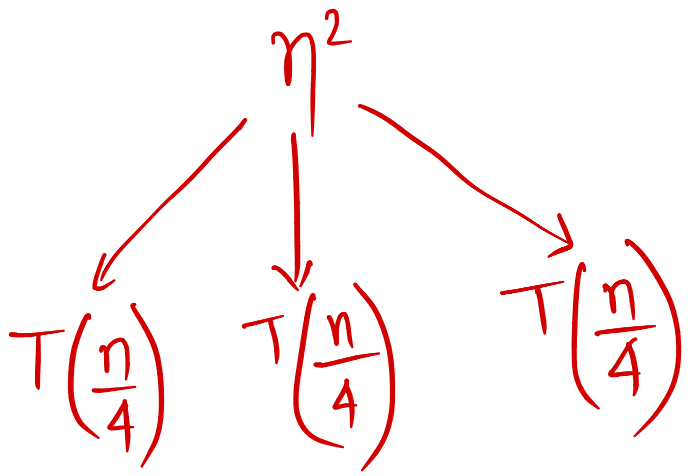
Similar to last case,

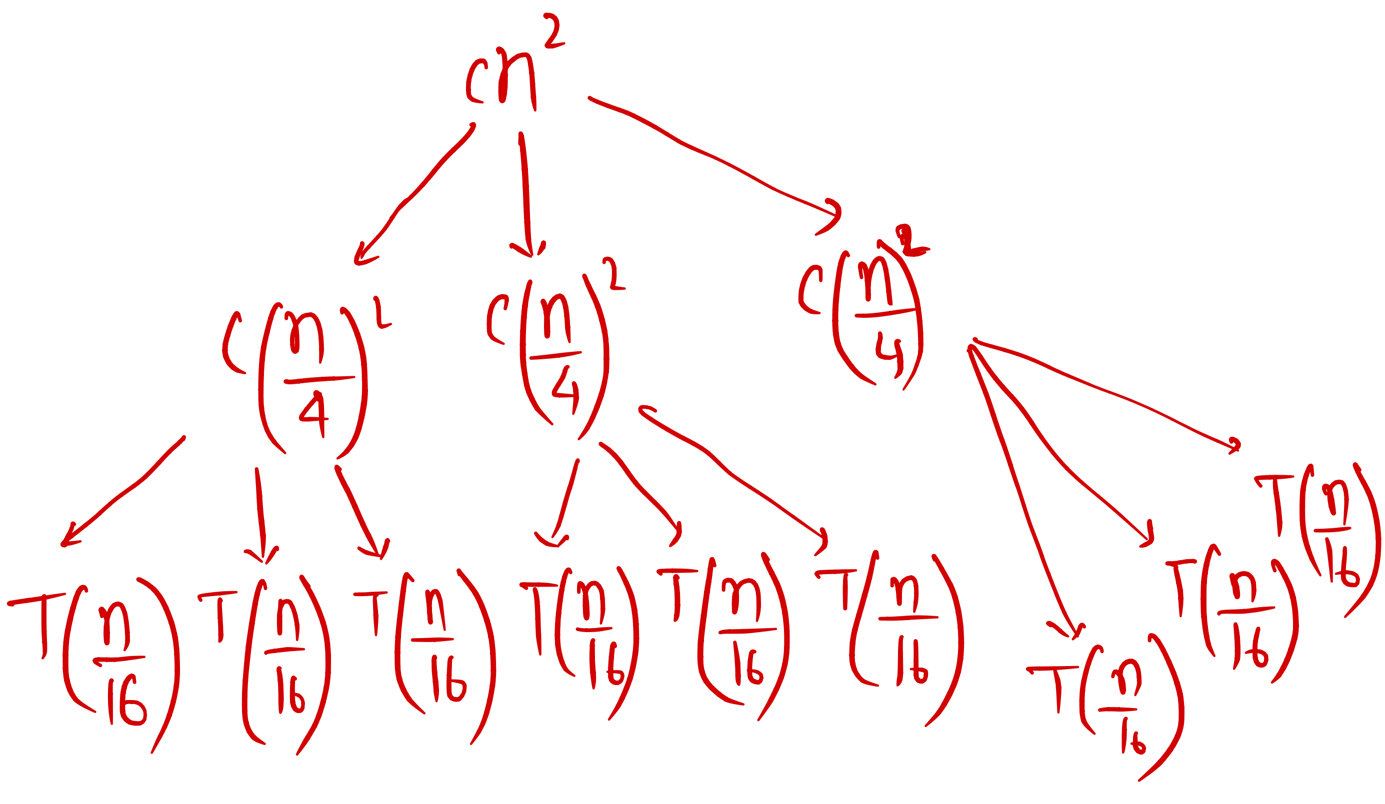
$$T(n) = \Theta(n)$$

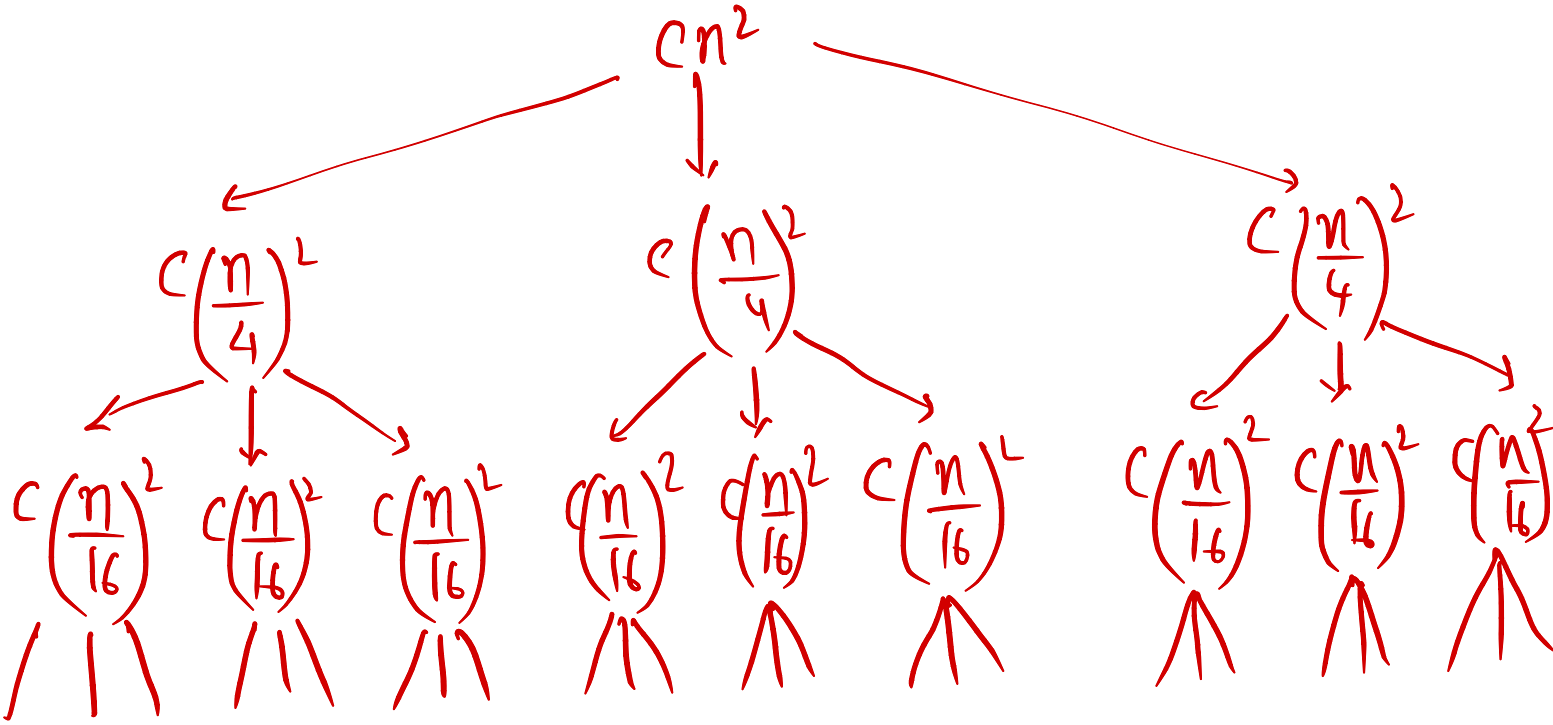
Example :-

$$T(n) = 3T\left(\frac{n}{4}\right) + \Theta(n^2)$$

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







Level	No. of Nodes	Size of Each Subproblem
0	3^0	$n/4^0$
1	3^1	$n/4^1$
2	3^2	$n/4^2$
3	3^3	$n/4^3$
⋮	⋮	⋮
$\log_4 n$	$3^{\log_4 n}$	1.

Level	No. of Nodes	Size of SBs	Cost of
0	3^0	$n/4^0$	cn^2
1	3^1	$n/4^1$	$c\left(\frac{n}{4}\right)^2$
2	3^2	$n/4^2$	$c\left(\frac{n}{4^2}\right)^2$
3	3^3	$n/4^3$	$c\left(\frac{n}{4^3}\right)^2$
⋮	⋮	⋮	⋮
4	$3^{\log_4 n}$	1.	c

$$T(n) = cn^2 + 3c\left(\frac{n}{4}\right)^2 + 3^2c\left(\frac{n}{4^2}\right)^2 + 3^3c\left(\frac{n}{4^3}\right)^2 + \cdots + 3^{\log_4 n-1}c\left(\frac{n}{4^{\log_4 n-1}}\right)^2 + c\,3^{\log_4 n}$$

$$= cn^2\left(1 + \frac{3}{4^2} + 3^2\cdot\frac{1}{4^4} + 3^3\cdot\frac{1}{4^6} + \cdots + 3^{\log_4 n-1}\cdot\frac{1}{4^{2\log_4 n-2}}\right) + c\,n^{\log_4 3}$$

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

$$+ c\, n^{\log_4 3}$$

$$= cn^2 \left(\left[1 - \left(\frac{3}{16}\right)^{\log_4 n}\right] \cdot \frac{1}{\left(1 - \frac{3}{16}\right)} \right) + cn^{\log_4 3}$$

$$= cn^2 \left(\frac{16}{13} \cdot \left[1 - \left(\frac{3}{16}\right)^{\log_4 n}\right] \right) + cn^{\log_4 3}$$

$$= cn^2 \left(\frac{16}{13} \cdot \left(1 - \frac{3^{\log_4 n}}{4^{\log_4 n^2}}\right)\right) + cn^{\log_4 3}$$

$$= cn^2 \left(\frac{16}{13} \cdot \left(1 - \frac{n^{\log_4 3}}{n^2}\right)\right) + cn^{\log_4 3}$$

$$= cn^2 \left(\frac{16}{13} \cdot \left(\frac{n^2 - n^{\log_4 3}}{n^2}\right)\right) + cn^{\log_4 3}$$

$$= c\cdot\left(\frac{16}{13}\right)\left(n^2 - n^{\log_4 3}\right) + c\,n^{\log_4 3}$$

$$T(n) = c\cdot\left(\frac{16}{13}\right)n^2 - c\left(\frac{16}{13}\right)n^{\log_4 3} + c\,n^{\log_4 3}$$

$$= \frac{16c}{13}n^2 - \frac{16c}{13}n^{\log_4 3} + c\,n^{\log_4 3}$$

$$= \frac{16c}{13}n^2 + \left(c - \frac{16c}{13}\right)n^{\log_4 3}$$

$$= \frac{16c}{13}n^2 - c'n^{\log_4 3} \qquad (c' > 0)$$

$$\frac{3}{13}c$$

$$\left.\begin{aligned} &T(n) \leq \frac{16c}{13}n^2 \\ &\text{and} \\ &T(n) \geq c\,n^2 \end{aligned}\right\} \Rightarrow T(n) \text{ is } \Theta(n^2)$$

Observation

Given a recurrence equation of form

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

Recall Cost table of $T(n) = 3T(n/4) + n$ using Recursion Tree method.

		Problem Size		
Levels	No. of Nodes	Cost/Node	Total Cost	
0	1, 1	n, $f(n)$	n	$f(n)$
1	3, a	$\frac{n}{4}$, $f\left(\frac{n}{b}\right)$	$\frac{3}{4}n$	$a f\left(\frac{n}{b}\right)$
2	3^2, a^2	$\frac{n}{4^2}$, $f\left(\frac{n}{b^2}\right)$	$\frac{3^2}{4^2}n$	$a^2 f\left(\frac{n}{b^2}\right)$
$\vdots$	$\vdots$	$\vdots$ $\vdots$	$\vdots$	$\vdots$
$\log_4 n - 1$	$3^{\log_4 n - 1}$, $a^{\log_b n - 1}$	$\frac{n}{4^{\log_4 n - 1}}$, $f\left(\frac{n}{b^{\log_b n - 1}}\right)$	$\left(\frac{3}{4}\right)^{\log_4 n - 1} n$	$a^{\log_b n - 1} f\left(\frac{n}{b^{\log_b n - 1}}\right)$
$\log_4 n$	$3^{\log_4 n}$, $a^{\log_b n}$	1, $f(1)$	$3^{\log_4 n}$	$a^{\log_b n} f(1)$

$$T(n) = \sum_{i=0}^{\log_b n - 1} a^i f\left(\frac{n}{b^i}\right) + a^{\log_b n} f(1)$$

Complexity ??

$$T(n) = \sum_{i=0}^{\log_b n - 1} a^i f\left(\frac{n}{b^i}\right) + n^{\log_b a} f(1)$$

So we have to discuss the complexity in three cases.

Case 1.

$$f(n) = O\left(n^{\log_b a - \varepsilon}\right) \quad (\varepsilon > 0)$$

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

Case 2

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

$$\left(T(n) = \Theta\left(n^{\log_b a} \cdot \log n\right)\right)$$

Case 3 :

$$f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right) \quad \left(T(n) = \Theta(f(n))\right)$$

(Master Theorem handles all the cases)

Master Theorem

(Proof of Master Theorem is not included but will be uploaded separately)

Let $a \geq 1$, $b > 1$ be constants, Let $f(n)$ be a function and let

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

where $T(n)$ be defined on non-negative integers by recurrence.

Then

(i) If $f(n) = O(n^{\log_b a - \varepsilon})$ for some $\epsilon > 0$,

Then, $T(n) = \Theta(n^{\log_b a})$

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

(iii) If $f(n) = \Omega(n^{\log_b a + \varepsilon})$, for some constant $\varepsilon > 0$, and if $af\left(\frac{n}{b}\right) \leq c f(n)$ for some $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$

Example :- (from CLRS)

$$T(n) = 5T\left(\frac{n}{2}\right) + \Theta(n^2)$$

$a = 5$ (≥ 1)

$b = 2$ (> 1)

$f(n) = n^2$

$$\log_2 5 = 2.3219$$

Relation between

$(f(n)=)$ n^2 and $n^{2.3219}$ $\left(= n^{\log_b a}\right)$

$$n^2 \text{ is } O\left(n^{(2.3219 - 0.3219)}\right)$$

(Here, ϵ is positive)

Hence,

Case I is applicable, hence $T(n) = \Theta\left(n^{\log_2 5}\right)$

(of Master Theorem)

Example 2

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

$a = 9 \ (\geq 1)$ ✓

$b = 3 \ (> 1)$ ✓

$$\log_3 9 = \log_3 3^2 = 2$$

Here, $f(n) = n$

Relation between n and $n^{\log_3 9} \ (= n^2)$

$$n \text{ is } O\left(n^{\log_3 9 - 1}\right)$$

Here, $\epsilon = 1$

So, Case I is applicable, hence

$$T(n) = \Theta\left(n^{\log_3 9}\right)$$

i.e. $T(n) = \Theta(n^2)$

Example 3:

$$T(n) = T\left(\frac{2n}{3}\right) + 1$$

$a = 1 \ (\geq 1)$ ✓

$b = \frac{3}{2} \ (> 1)$ ✓

$$f(n) = 1 = \Theta(1)$$

Growth rate

$$n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1$$

$$\left(\log_{3/2} 1 = \frac{\log_e 1}{\log_e 3/2} = 0 \right)$$

$$f(n) = \Theta\left(n^{\log_{3/2} 1}\right) = \Theta(1)$$

Hence, $T(n)$ is $\Theta(1 \cdot \log n)$

Example : 4 $T(n) = 3T\left(\frac{n}{4}\right) + n\log n$

$a = 3 (\geq 1)$ ✓

$b = 4 (> 1)$ ✓

$f(n) = n\log n$

$$n^{\log_4 3} = n^{0.79}$$

$$n\log n \geq c \cdot n^{0.79 + 0.20}$$

$$f(n) = \Omega\left(n^{\log_4 3 + \varepsilon}\right)$$

For sufficiently large n,

$$a f\left(\frac{n}{b}\right) = 3 f\left(\frac{n}{4}\right) = 3 \frac{n}{4} \log\left(\frac{n}{4}\right)$$

$$\leq 3/4 \, n \log n$$

Hence, Case III applies.

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

Example 5.

$$T(n) = 2T(\sqrt{n}) + \log_2 n$$

Let:

$$2^k = n$$

$$T(2^k) = 2T(2^{k/2}) + \log_2(2^k)$$

$$T(2^k) = S(k) \quad (\text{let})$$

Then, $T(2^{k/2}) = S(k/2)$

$$S(k) = 2S(k/2) + \log_2(2^k)$$

$$S(k) = 2S(k/2) + k$$

$$\begin{pmatrix} a = 2 \\ b = 2 \end{pmatrix} \qquad k^{\log_2 2} = k$$

$$k = \Theta(k^{\log_2 2})$$

Hence, Case II is applicable, Hence

$$S(k) = \Theta(k \cdot \log_2 k)$$

$$S(k) = \Theta(k \cdot \log_2 k)$$

$$S(k) = T(2^k) = T(n)$$

$$T(n) = \Theta(2^k \log_2 2^k)$$

$$= \Theta(n \log_2 n)$$

Example 6

$$T(n) = T(\sqrt{n}) + \log_2 n$$

$n = 2^k$ and $S(k) = T(2^k)$

$$S(k) = S(k/2) + \log_2(2^k)$$

$$S(k) = S(k/2) + k$$

$a = 1$

$b = 2$ $\qquad k^{\log_2 1} = k^0 = 1$

$$f(k) = k \quad \text{is} \quad \Omega(k^{\log_2 1 + 1})$$

Further,

$$a f\left(\frac{k}{b}\right) = 1 \cdot f\left(\frac{k}{2}\right) \leq \frac{k}{2} \quad (\forall k)$$

Hence, condition is satisfied.

$$S(k) = \Theta(k)$$

$$S(k) = T(2^k) = T(n) = \Theta(\log_2 n)$$

$$T(n) = \Theta(\log_2 n)$$

Example 7

$$T(n) = T(\sqrt{n}) + c$$

$n = 2^k$ and $T(2^k) = S(k)$

$$S(k) = S(k/2) + c$$

$$k^{\log_2 1} = k^0 = 1$$

Hence, $f(k) = \Theta(k^{\log_2 1}) = \Theta(1)$

Case II is applicable :-

$$S(k) = \Theta(k^{\log_2 1} \cdot \log k)$$

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