$C.2.\ Recurrence\ Tree\ Method - Example\ 2$

$$t_n = \begin{cases} 1 & for\ n = 1 \\ t_{n-1} + a & for\ n > 1 \end{cases}$$

$Using\ the\ recurrence - tree\ method, determine\ the\ final\ asymptotic\ complexity\ of\ the\ tree\ for\ a = 1\ and\ a = n.$

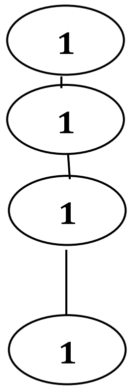
<u>$Solution$:</u>

$When\ a = 1\ the\ recurrence\ equation\ is\ as\ follows:$

$$t_n = \begin{cases} 1 & for\ n = 1 \\ t_{n-1} + 1 & for\ n > 1 \end{cases}$$

Therefore, the recurrence tree for this recurrence equation would be shown below:

Level	No. of problems	Problem Size	Work done = Problem Size $\times$ No. of Problems
0	1	1	$1 \times 1 = 1$
1	1	1	$1 \times 1 = 1$
2	1	1	$1 \times 1 = 1$
. .	. .	. .	. .
$n-1$	1	1	$1 \times 1 = 1$



One can observe that the work done at every level is 1 and the size of the problem is reduced by a factor of 1 at every level .

Therefore, at the level $n-1$, the problem size and work done would be 1. The total number of levels is n (as the level of the root is 0). Therefore, the final cost can be estimated as follows:

$$Total\ Cost = \sum_{i=1}^{n} 1 = 1 + 1 + 1 + \cdots n\ times = 1 \times n = n.$$

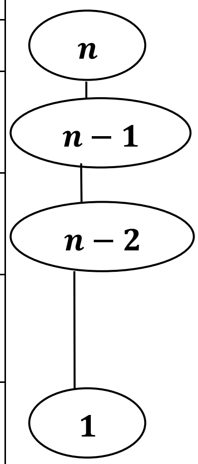
$Therefore, the\ asymptotic\ complexity\ would\ be\ \Theta(n).$
$Problems\ such\ as\ `Towers\ of\ Hanoi`\ and\ `Handshake\ Problem`\ can\ be\ expressed\ using\ this\ type\ of\ a\ recurrence\ equation.$

$When\ a = n, the\ recurrence\ equation\ would\ be\ as\ follows:$

$$t_n = \begin{cases} 1 & for\ n = 1 \\ t_{n-1} + n & for\ n > 1 \end{cases}$$

$The\ recurrence\ tree\ is\ shown\ below:$

Level	No. of problems	Problem Size	Work done = Problem Size $\times$ No. of Problems
$\mathbf{0}$	$\mathbf{1}$	$\boldsymbol{n}$	$\mathbf{1} \times \boldsymbol{n} = \boldsymbol{n}$
$\mathbf{1}$	$\mathbf{1}$	$\boldsymbol{n} - \mathbf{1}$	$\mathbf{1} \times (\boldsymbol{n} - \mathbf{1}) = \boldsymbol{n} - \mathbf{1}$
$\mathbf{2}$	$\mathbf{1}$	$\boldsymbol{n} - \mathbf{2}$	$\mathbf{1} \times (\boldsymbol{n} - \mathbf{2}) = \boldsymbol{n} - \mathbf{2}$
. .	. .	. .	. .
$\boldsymbol{n} - \mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1} \times \mathbf{1} = \mathbf{1}$



We can observe that at entry level the work done is reduced by 1 from previous term from total `n` terms.

The work done is given as: $\boldsymbol{n} + (\boldsymbol{n} - \mathbf{1}) + (\boldsymbol{n} - \mathbf{2}) + \cdots + \mathbf{1}$.

The total number of levels is $\boldsymbol{n}$ ***(from 0 to*** $\boldsymbol{n} - \mathbf{1}$***).***

As there are `n` levels, the total cost can be estimated as follows:

$$\boldsymbol{Total\ Cost = \sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \Theta(n^2)}$$

****