Factorial Time complexity using Recursion Tree Method

$$T(n) = \begin{cases} 1 & , for\ (n = 0) \\ 1 & , for\ (n = 1) \\ T(n-1) + T(1) & , for\ (n > 1) \end{cases}$$

Or for

$$T(n) = \begin{cases} 1 & , for\ (n = 0) \\ T(n-1) + 1 & , for\ (n > 0) \end{cases}$$

We know :

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

When a is 1 *, problem size becomes* 1 *i.e. constant.*

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

Level	No. of problems	Problem Size	Work done = Problem Size × 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 $O(n)$.

******