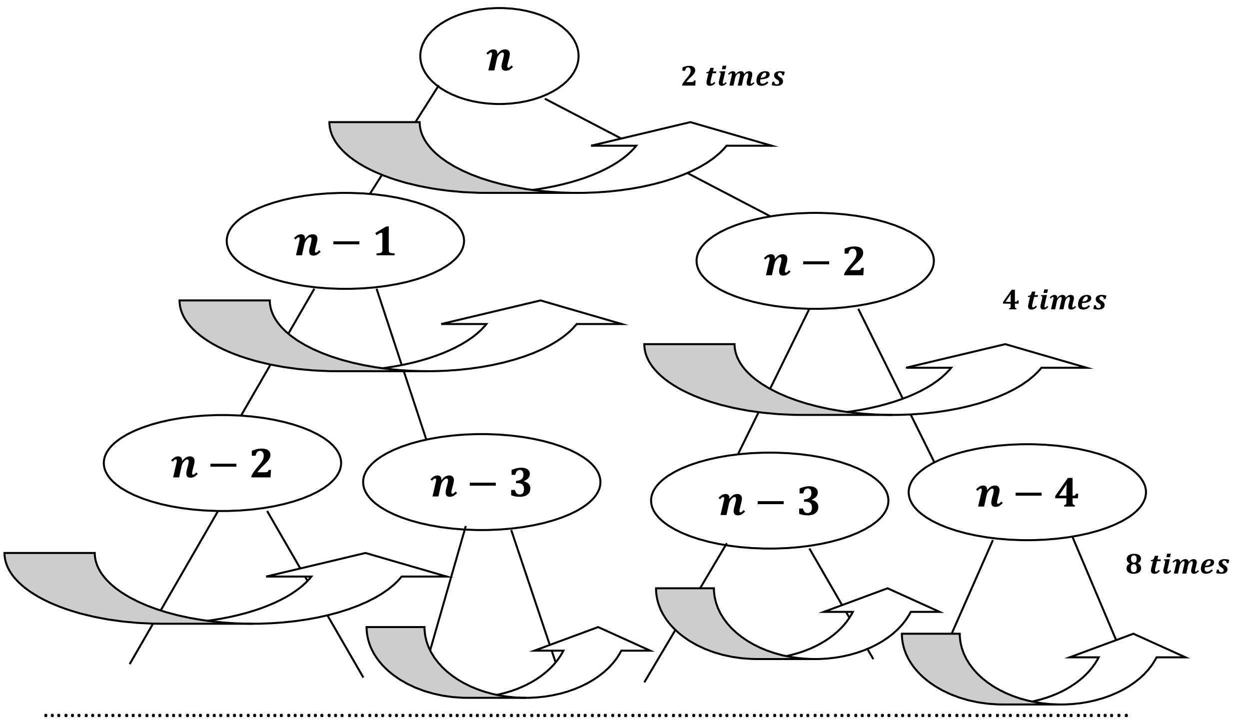
$Fibonacci\ Series\ Time\ Complexity$
$with\ Recursion\ Tree$

$In\ fibonacci\ series : T(n) = T(n-1) + T(n-2), where\ `n`\ is\ greater\ than\ 1$, $when\ `n` = 0 , T(n) = 0\ and\ T(n) = 1 , when\ `n` = 1, these\ are\ base\ cases\ for$ $fibonacci\ series.$

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

$To\ start\ with: -$

$If\ the\ recursion\ goes\ from\ `n`\ to\ `1`\ times, considering\ every\ nodes, we\ get:$



$Level$	$No.of$ $problems$	$Problem$ $size$	$Work$ $Done$
0	1	1	1
1	2	1	2
2	4	1	4
.	.	.	.
.	.	.	.
$n-1$	2^n	1	2^n

$As\ we\ know$:

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

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

$As\ we\ have\ same\ growth\ \ shown\ in\ the\ recurrence$

$tree\ for\ T(n-1) and\ T(n-2)\ \ we\ have\ assumed\ :$

$T(n-1) \approx T(n-2)\ for\ previous\ problem\ .$

$Now\ we\ can\ see\ the\ gowth = 1 + 2 + 4 + \cdots 2^{n-1}$

$We\ can\ rewrite\ it\ as$

$\boldsymbol{if\ we\ see\ it, 1 + 2 + 2^2 + \cdots is\ in\ geometric\ finite\ series}$:

$$\Longrightarrow \frac{x^{n+1} - 1}{x - 1} = \frac{2^{n+1} - 1}{2 - 1} = 2^{n+1} - 1 = O(2^n - 1) = O(2^n)\ is\ time$$

$\boldsymbol{complexity}$.

****