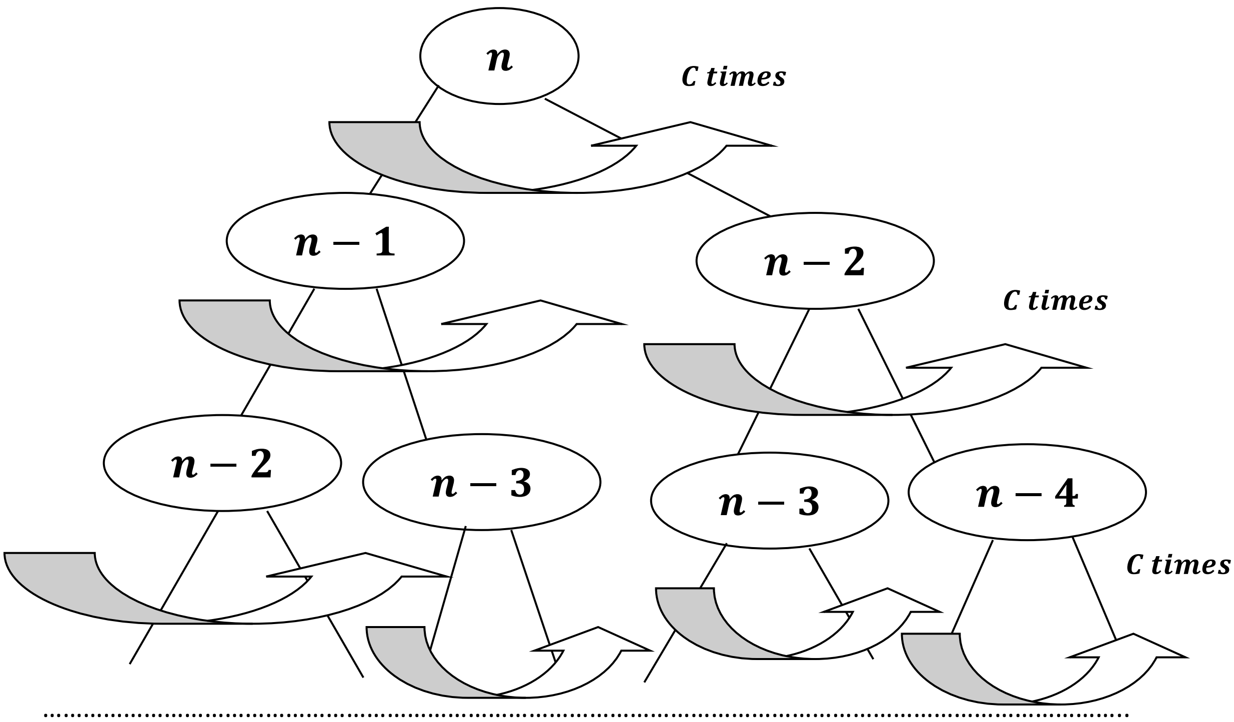
$\textit{Fibonacci Series Time Complexity}$
$\textit{with Recursion Tree}$

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

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

$\textit{To start with}: -$

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



$Level$	$No. of$ $problems$	$Problem$ $size$	$Work$ $Done$
0	c	1	c
1	c	2	$2c$
2	c	4	$4c$
.	.	.	.
.	.	.	.
$n-1$	c	2^{n-1}	$c \times 2^{n-1}$

$Here\ `c`\ is\ constant.$

$Hence\ we\ have\ assumed : T(n-1) \approx T(n-2)\ for$

$previous\ problem\ , because\ of\ the\ same\ growth\ as\ shown$

$in\ recurrence\ tree,$

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

$We\ can\ rewrite\ it\ as$

$: 2^0 \times c + 2^1 \times c + 2^2 \times c + \cdots + c \times 2^{n-1}$

$\Rightarrow c\{1 + 2 + 2^2 + .. + 2^{n-1}\}$

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

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

$complexity.$

****