Fibonacci Series Time Complexity with Backward Substitution Method

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***.

To start with: $-$

$T(n) = T(n-1) + T(n-2) + 1$, ***here*** 1 ***is constant for the base case***.

Assuming, $T(n-1) \approx T(n-2)$, ***we get*** :

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

By Backward Substitution:

Substitution the values of $T(n-1)$ ***in the recurrence equation, one gets the following equations***:

$$= 2 \times [2 \times T(n-1-1) + 1] + 1$$

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

Substitution the values of $T(n-2)$ ***in the recurrence equation, one gets the following equations***:

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

$= 8T(n-3)+7$

Substitution the values of $T(n-3)$ ***in the recurrence equation, one gets the following equations***:

$= \ 2[8T(n-3-1)+7\ ]+1$

$= \ 16\,T(n-4)+15$

Hence by repeating the process , one can observe that at the ith iteration, this equation would be as follows:

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

If $i = n-\ 1$ ***, we will have***:

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

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

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

$= 2^{n-1}\times 1+2^{n-1}-1$ [$We\ know, T(1)=1$]

$= 2^{n-1}+2^{n-1}-1$

$= 2 \times (2^{n-1}) - 1$ [$As, 2^{n-1} + 2^{n-1} = 2 \times (2^{n-1})$]

$= 2^{n-1+1} - 1$

$= 2^n - 1$

$= O(2^n - 1)$

$= O(2^n)$

$Therefore, O(2^n)$ *is the answer*.

****