Complexity of Fibonacci number

Fibonacci numbers are generated by:

F(n) = F(n-1) + F(n-2).

i.e. 0, 1, 1, 2, 3, 5, 8, 13, 21, ………………….

The characteristic equation to generate this sequence is $x^n = x^{n-1} + x^{n-2}$.

Or dividing x^{n-2} , it is $x^2 = x + 1$.

So $x^2 - x - 1 = 0$.

It has 2 solution: $\frac{1\pm\sqrt{1+4}}{2}\ i.e.\frac{1+\sqrt{5}}{2}\ ,\frac{1-\sqrt{5}}{2}$.

So nth fibonacci number can be written as $F_n = \left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)^n$.

So complexity of fibonacci number generating is $O(F_n) = O\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right) + O\left(\left(\frac{1-\sqrt{5}}{2}\right)^n\right)$

So $O(F_n) = O\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$, which is simplified as $(1.6180)^n \approx 2^n$.

So time complexity of fibonacci i.e. time required to calculate nth order fibonacci number is $O(2^n)$. .