Implementation-1:

from the code we can write:

$$1 = 2^0$$

T(n)

T(n-1)   T(n-2) $2 = 2^1$

T(n-1)   T(n-2)   T(n-3)   T(n-4) $4 = 2^2$

T(n-2)   T(n-3)   T(n-3) T(n-4)   T(n-4) T(n-5)   T(n-5)   T(n-6)

$$8 = 2^3$$

Now,

$$2^0 + 2^1 + 2^2 + 2^3 + \cdots \cdots 2^n$$

$$T(n) = \sum_{0}^{n} 2^i$$
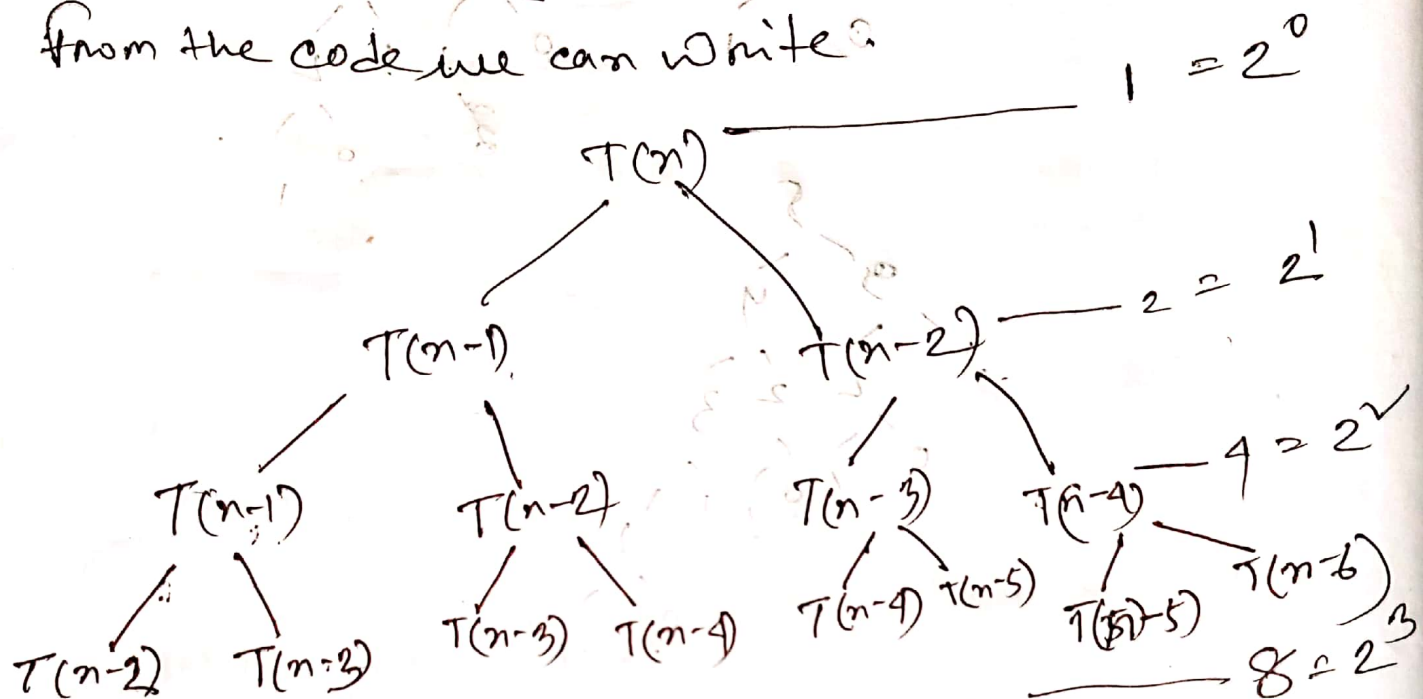
$$= \frac{2^{n+1} - 1}{2 - 1}$$

$$= \frac{2^n - 1}{1}$$

So the time complexity is $O(2^n)$ for this code.

Implementation - 2

As we can see in this code memoization is been used so there will be no. repeation of the calculation for the same value of n. So we can say that the time complexity for this code will be

$O(n)$.

In this code we can See that there are total 3 nested for loops. And every for loop is iterating n times @ So the time complexity of this code will be $O(n^3)$.