For the implementation 1 the function is using recursion

So let,

$$T(n) = T(n-1) + T(n-2) + c$$

$$T(0) = T(1) = 1$$

$$T(n-2) \approx T(n-1)$$

$$T(n) = 2T(n-1) + c$$

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

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

$$\vdots \quad \vdots \quad \vdots$$

$$= 2^k \cdot T(n-k) + (2^k - 1)c \longrightarrow (i)$$

Now

$$n - k = 0$$

$$n = k$$

$$k = n$$

eq(i) becomes

$$T(n) = 2^n T(0) + (2^n - 1)c$$

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

$$T(n) \simeq 2^n$$

So, for the implementation 1 the time complexity is $O(2^n)$.

This function is exponential function and have a bad time complexity.

The implementation 2 is using simple loop / for loop & memoization method.

Let's take time complexity of n to be T(n)

hence $T(n) = T(n-1) + T(n-2)$.

This is because $T(n-2)$ is in the cache when we calculate $T(n-1)$, so the operation of $T(n-1)$ is 1, so $T(n) = T(n-1) + 1$

$= T(n-2) + 2 = T(n-n) + n$.

And $T(0)$ is 1, so

$$T(n) = O(n+1)$$

$$T(n) = O(n)$$

This is linear function so that why impelementation 2 has bethr time complexity then impelimentation 1.