

In Implementation 1 recursive method is used.

So Let,

$$\begin{aligned}T(n) &= 2T(n-1) + C \\&= 4T(n-2) + 3C \\&= 8T(n-3) + 7C \\&= 2^k T(n-k) + (2^k - 1)C \dots \textcircled{1}\end{aligned}$$

Now,

$$n - k = 0$$

$$\therefore n = k \dots \textcircled{II}$$

Using eqn \textcircled{I} and \textcircled{II} ,

$$\begin{aligned}T(n) &= 2^n T(0) + (2^n - 1)C \\&= 2^n(1 + C) - C \\&\approx 2^n\end{aligned}$$

\therefore for Implementation -1 time complexity is $O(2^n)$

In Implementation-2 there is only one "for loop" used.

So time complexity is $O(n)$

As $O(2^n)$ is an exponential function and $O(n)$ is a linear function, Implementation-2 is better than Implementation-1

$$(1) \dots O(1-x^n) + (n-x)T^n =$$

$$(11) \dots x = n$$

$$O(1-x^n) + (0)T^n = (n)T$$

$$O - (O+1)^n =$$

$$^n \approx$$

$(^n S) O$ is time complexity of Implementation-1