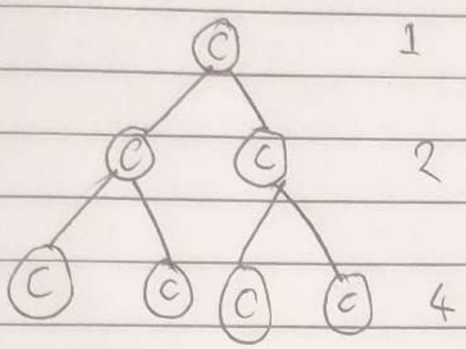
Name: Sadnan Nafis
ID: 21201249

Problem 2a:

Implementation 1 uses recursion to find the fibonacchi number, and it calculates the same fibonacchi number multiple times if required.

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



$1 + 2 + 4 + \dots + 2^n$

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

$= O(2^n)$ [Time complexity of implementation 1]

Implementation 2 uses memoization technique to store fibonacchi numbers that has also been calculated. So the repeated sub-trees don't have to be calculated using recursion again, thus saving time.

Time complexity of implementation $= O(n)$ (linear)

$\therefore$ Implementation 2 is far better than implementation 1. The graph in problem 2b also highlights this time complexity.