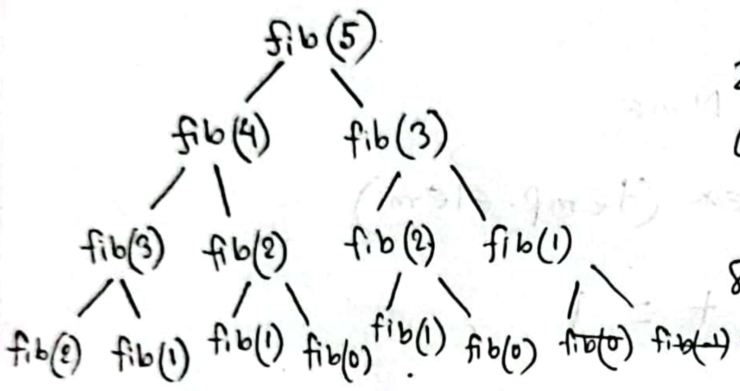
Task 2(a)

Imple mentation 1

Let $n = 5$



$1 \rightarrow 2^0$
$2 \rightarrow 2^1$
$4 \rightarrow 2^2$
$8 \rightarrow 2^3$

Level $= 4 = n-1$

$\therefore 2$

we can see, at each level, the number ~~that~~ to call the function increases by ~~Double~~ $2^{(n-1)}$. (Ignoring fib(1) at level 3)

$\therefore$ Time complexity $= O(2^{n-1}) = O(2^n \cdot 2^{-1}) = O(2^n)$

(Ans)

Implementation 2:

The first 8 lines time complexity is $T(1)$ or $T(0)$ depending on the conditions.

9th line run for (2 to n):

$= T(n-1)$

last line will run once $(T(1))$

So, total time complexity $= O(c + n - 1)$

c is a constant that is sum of $T(1)$ & $T(0)$

$= O(n)$

(Ans)