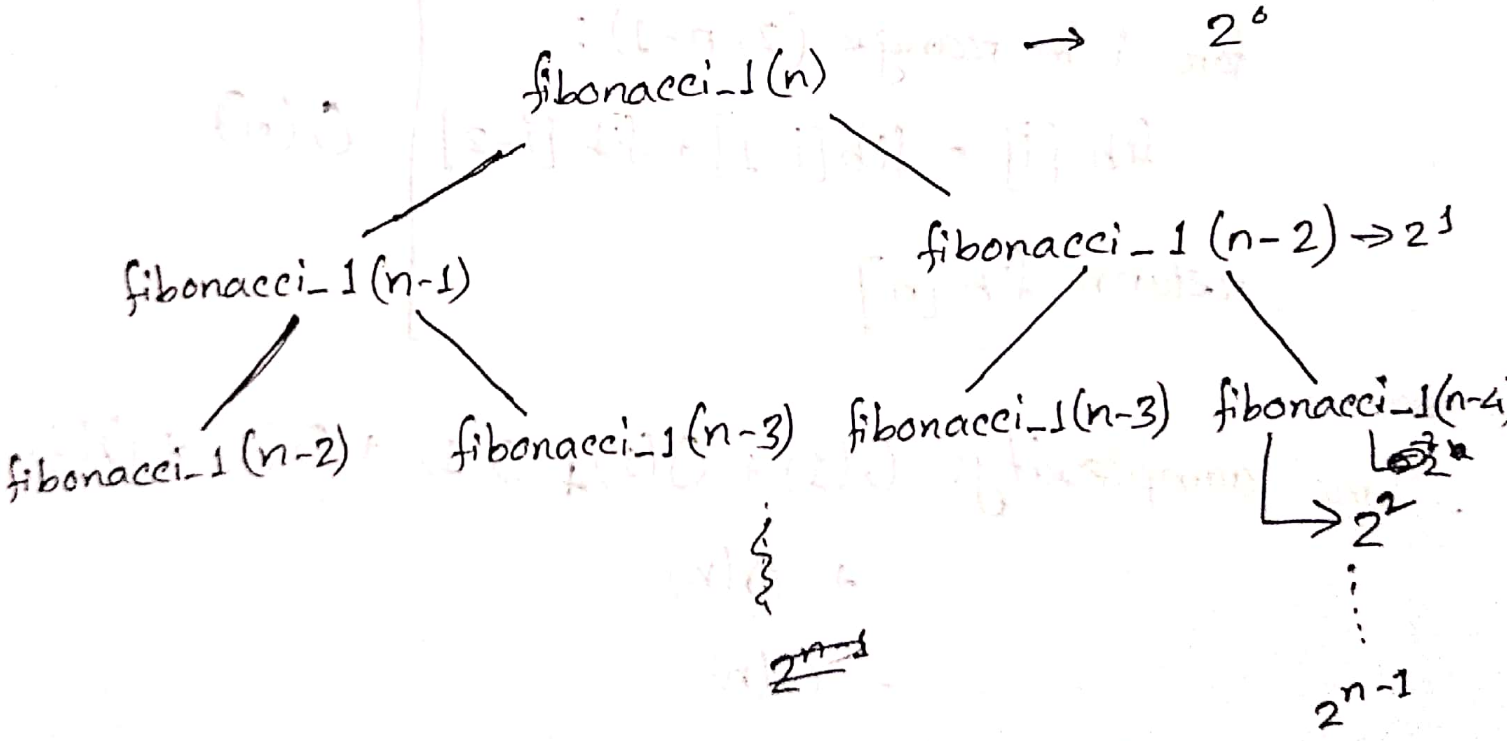
Task - 2(a)

Implementation - 1

```
def fibonacci_1(n):
    if n<0:
        print("Invalid input!")          ] → O(1)

    elif n<=1:
        return n                         ] → O(1)

    else:
        return fibonacci_1(n-1)+fibonacci_1(n-2)   ] O(2^{n-1})
```



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

$= O(2^n)$

Implementation - 2:

```
def fibonacci_2 (n) :
    if n<0 :
        return "Invalid Input"      ] O(1)

    if n<=1 :
        return n                    ] O(1)

    fib = [0]* (n+1)] O(1)
    fib [0] = 0   ] O(1)
    fib [1] = 1   ] O(1)
    for i in range (2, n+1) :
        fib [i] = fib[i-1] + fib [i-2]   ] O(n)

    return fib [n]
```

$\therefore$ Time complexity: $O(1) + O(1) + O(1) + O(1) + O(1)$

$+ O(n)$

$= O(n)$

So, from the time complexity we can tell that the implementation 2 is faster.