Subject: Refatul Islam 20101482 Date:

Answer-02(i)

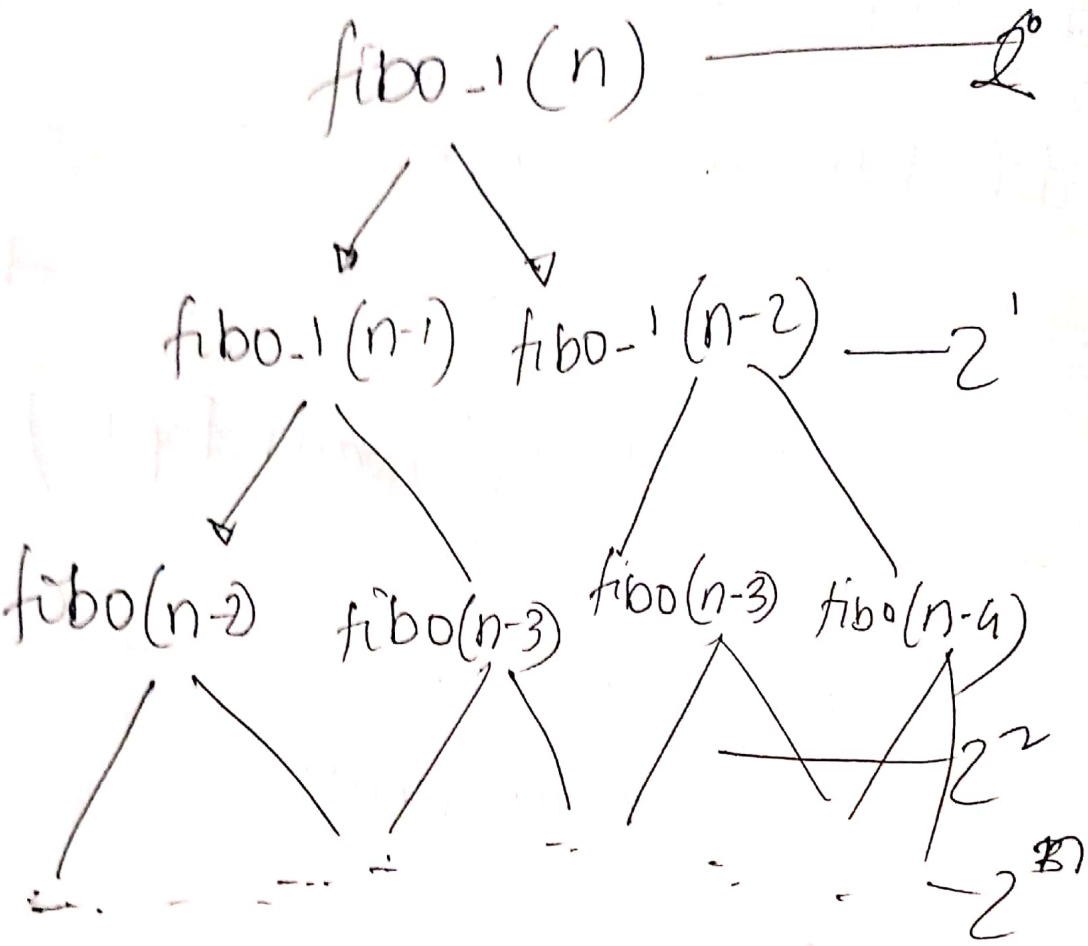
```
def fibonacci_1(n):                                  time com
    if n<=0:  ------------------------------------> O(1)
        print("Invalid Input")

    else if n<=2:  ------------------------------- O(1)
        return (n-1)

    else.
        return fibonacci_1(n-1) +
                        fibonacci(n-2)
```

tree diagram.

Subject : ..... Date : .....



Combining the all the steps from above, we get

$= O(1) + O(1) + 2^0 + 2^1 + 2^2 + \cdots 2^n$

$= O(1) + O(1) + O(2^{n+1} - 1)$

Subject : ..... Date : .....

$= O(2^{n+1} - 1)$ [ignoring the constants and lower values.]

$= O(2^n)$

therefore, the time complexity

$\longrightarrow O(2^n)$

Ans..

Subject : ..... Date : .....

2nd, Algorithm.

```
def fibonacci-2(n):
    array = [0, 1]                          ——> O(1)
    if (n<0) then,                          ——> O(1)
        print ("invalid input")
    else if (n<=2)then,                     ——> O(1)
        array[n-1]
    return fibonacci.

    else:
        for i in rang(2, n):                ——> O(n)
            array.appends(array[i-1] +
                          array[i-2])
```

Computing all the steps, we get.

$O(n) + O(1) + O(1) + O(1)$

$= O(n)$. [ignoring all constants and lower values]

$\therefore O(n)$

Time complexity = $O(n)$

Ans